

Tight Bounds for Low Dimensional Star Stencils in the External Memory Model

Philipp Hupp and Riko Jacob

Institute of Theoretical Computer Science, ETH Zürich, Zürich, Switzerland

Abstract. Stencil computations on low dimensional grids are kernels of many scientific applications including finite difference methods used to solve partial differential equations. On typical modern computer architectures such stencil computations are limited by the performance of the memory subsystem, namely by the bandwidth between main memory and the cache. This work considers the computation of star stencils, like the 5-point and 7-point stencil, in the external memory model. The analysis focuses on the constant of the leading term of the non-compulsory I/Os. Optimizing stencil computations is an active field of research, but so far, there has been a significant gap between the lower bounds and the performance of the algorithms. In two dimensions, matching constants for lower and upper bounds are provided closing a gap of 4. In three dimensions, the bounds match up to a factor of $\sqrt{2}$ improving the known results by $2\sqrt{3}\sqrt{B}$, where B is the block (cache line) size of the external memory model. For higher dimensions n , the presented lower bounds improve the previously known by a factor between 4 and 6 leaving a gap of ${}^n\sqrt{n!} \approx \frac{n}{e}$.

1 Introduction

Solving Partial Differential Equations (PDEs) is one of the most common tasks in scientific computing. A standard way to discretize low dimensional Euclidean spaces for these computations are regular grids. Applying a finite difference method on this discretization turns a differential operator into a linear function of a grid point and its neighbors. Consider, for example, the linear approximation of the Laplacian on a regular two dimensional grid as given by $\Delta u(x, y) \doteq \frac{1}{h^2} [u((x-h), y) + u((x+h), y) + u(x, y-h) + u(x, y+h) - 4u(x, y)]$. This defines the so called 5-point or 1-star stencil and turns the differential equation at hand into a very regular sparse system of linear equations. To make use of the sparsity, such systems are typically solved with iterative solvers like the Jacobi or Gauss-Seidel method. The kernel of these methods is the evaluation of the underlying stencil, making it the most performance critical component of the computation.

One characteristic of the stencil computation is that it performs relatively few floating point operations on the data. Frequently, the theoretically available peak floating point performance cannot be achieved because the memory system is the bottleneck limiting the speed. Over the last decades the performance of the

memory systems did not increase at the rate of the CPU performance. Hence, optimizing the memory access became the main focus when designing high performance code in these situations. This focus is reflected in the computational models used to analyze algorithms and complexities of computational tasks. In this paper we work with the I/O-model that reflects two arbitrary levels of the memory hierarchy. In this model the complexity is given by the number of I/O operations, counting the number of cache misses or respectively the number of page faults. The model has the parameters M (main memory size) and B (block size). M describes the number of variables fitting into the memory in which computations are performed and B the size of a block (in variables) in which the disk is organized. These parameters reflect the two levels of the memory hierarchy that the model focuses on.

Stencil operations are not only easy in the sense that there is hardly any choice in the floating point operations. The majority of the I/O operations is already needed for reading the input and writing the output. In fact, many simple algorithms for the 5-point stencil are within a factor of 5 of this lower bound and the classical asymptotic analysis is too coarse to give interesting insights. I/O operations related to the initial read of the input and the final write of the output are called *compulsory* I/Os or *cold* cache misses. All other I/Os are called *non-compulsory* I/Os or *capacity* misses (because they are unnecessary for sufficiently large M). The analysis of stencil operations carried out in this paper gives almost matching bounds in the sense that it focuses on the constant of the leading term in the asymptotics of the non-compulsory I/Os.

Here, the lower bounds did not only provide part of the complexity results, but actually guided the construction of the data layouts and algorithms improving the upper bounds. What remains open is, on one hand, an experimental consideration how to turn the proposed algorithms into high performance code, and, on the other hand, to find matching bounds for situations with non-trivial B and standard layouts like a usual row or column layout. For the latter problem, we expect that the lower bound can be strengthened.

1.1 Problem Definition

The computational model is an I/O Model similar to [1] and [2]. There are two levels of memory, an *external memory* of infinite size on which all data is stored initially and an *internal memory* of size M to which the data has to be loaded to perform computations. The external memory is organized in *blocks* of size B . We classify the I/Os into cold misses or *compulsory I/Os*, which account for the first access to an element and writing the final output, and capacity misses or *non-compulsory I/Os* which are caused by the limited size of the internal memory.

The *computation graphs* consist of two layers, an input and an output layer, of either the n dimensional *grid* or *torus*. $[k]$ abbreviates $\{1, \dots, k\}$. $[k_1] \times [k_2] \times \dots \times [k_n]$ denotes the n -dimensional grid and $\mathbb{Z}_{k_1} \times \mathbb{Z}_{k_2} \times \dots \times \mathbb{Z}_{k_n}$ the n -dimensional torus of side lengths k_i . Two vertices x and y are joined if and only if $\|x - y\|_1 = 1$ where the component-wise computations are done in \mathbb{Z} for the

grid and in \mathbb{Z}_{k_i} for the i .th component of the torus. Denote with V the vertices of either the grid or torus.

The stencils considered are called *star stencils*. In n dimensions the s -star stencil S_s with center x is defined as $S_s(x) := \{y \in G : \|y - x\|_1 \leq s\}$. The most common star stencil is the 1-star stencil. Upper (lower) bounds for the s -star stencil induce upper (lower) bounds for all stencils which are subsets (supersets) of the s -star stencil. Hence meaningful choices also include $s = 2$ and $s = 3$. Therefore s is assumed to be a small constant in our discussion. The computation graph for S_s consists of two copies of V , $V_{in} := V \times \{\text{in}\}$ and $V_{out} := V \times \{\text{out}\}$, the vertices of which are connected in the following way: $E := \{(v_{in}, v_{out}) \in V_{in} \times V_{out} : \|v_{in} - v_{out}\|_1 \leq s\}$. The norm is computed as if both vertices would belong to V . The computation graph is then $(V_{in} \sqcup V_{out}, E)$. Denote by x_{in} and x_{out} the corresponding vertices of the input and output layer.

Evaluating S_s at $y_{out} \in V_{out}$ requires that all vertices to which y_{out} is connected, namely $S_s(y_{in})$, are in internal memory, i.e. we do not allow partial computations. The goal is to evaluate S_s for all vertices of V_{out} and write the results to external memory.

1.2 Results

This work examines the leading term of the non-compulsory I/Os of the s -point stencil. In two dimensions, matching lower and upper bounds are given closing a multiplicative gap of 4. In three dimensions the provided bounds match up to a factor of $\sqrt{2}$ improving the known results by a factor of $2\sqrt{3}\sqrt{B}$. For dimensions bigger than three, the lower bounds are improved between a factor of 4 and 6 leaving a gap of $\sqrt[n-1]{n!} \approx \frac{n}{e}$ for higher dimensions n .

We assume that the grid sides are ordered, $k_1 \geq \dots \geq k_n$, and significantly larger than the internal memory, namely $k_1, k_2 \geq 2nM + M + 1$ and $k_i \geq 2nM + 1$ for $i \in \{3, \dots, n\}$. The asymptotics are considered for $k_i \rightarrow \infty$, $M \rightarrow \infty$ and $B \rightarrow \infty$ while assuming $\frac{k_n}{M} \rightarrow \infty$ and $\frac{M}{B} \rightarrow \infty$. Denote by $C_s(k_1, \dots, k_n)$ the number of I/Os to evaluate the s -point stencil on $[k_1] \times \dots \times [k_n]$. Then the following holds (assuming n and s are constant):

$$\begin{aligned} C(k_1, k_2) &= \left(2 + 4 \frac{s^2}{M} \cdot \left\{ \begin{array}{l} 1 + \mathcal{O}\left(\frac{B}{M} + \frac{M}{k_1}\right) \\ 1 - \mathcal{O}\left(\frac{1}{M} + \frac{M}{k_2}\right) \end{array} \right\} \right) \cdot \frac{k_1 k_2}{B} \\ C(k_1, k_2, k_3) &= \left(2 + \frac{8}{\sqrt{3}} \cdot \frac{s^{3/2}}{\sqrt{M}} \cdot \left\{ \begin{array}{l} \sqrt{2} + \mathcal{O}\left(\sqrt{\frac{B}{M}}\right) \\ 1 - \mathcal{O}\left(\frac{1}{\sqrt{M}} + \frac{\sqrt{M}}{k_3}\right) \end{array} \right\} \right) \cdot \frac{k_1 k_2 k_3}{B} \\ C(k_1, \dots, k_n) &= \left(2 + 4 \cdot 2^{1/(n-1)} \cdot (n-1) \cdot \sqrt[n-1]{\frac{s^n}{M}} \cdot \left\{ \begin{array}{l} 1 + \mathcal{O}\left(\sqrt[n-1]{\frac{B}{M}}\right) \\ \frac{1}{\sqrt[n-1]{n!}} - \mathcal{O}\left(\frac{1}{\sqrt[n-1]{M}} + \frac{\sqrt[n-1]{M}}{k_n}\right) \end{array} \right\} \right) \cdot \frac{\prod_{i=1}^n k_i}{B} . \end{aligned}$$

The bounds consist of three parts. The first part is the constant 2 accounting for the compulsory I/Os. The second part is the leading term of the non-compulsory I/Os on which this work focuses. The third part characterizes lower order terms that we do not explore further.

Both lower bounds and algorithms can be transferred to parallel external memory (PEM) as introduced in [3], as long as the number P of processors is smaller than $\frac{1}{M} \prod_{i=1}^{n-1} k_i$. In this case, the complexities are reduced by a factor of P . Unlike with classical computational complexity (i.e. on a PRAM), there cannot be a general simulation of a parallel algorithm on a single processor that is only P times slower (it is possible to make use of the combined internal memory of size $P \cdot M$). Still, the lower bounds in this paper work in the parallel setting just as well: One round of the parallel computations, as defined by a certain number of non-compulsory I/Os, cannot evaluate more stencils than its serial counterpart. Regarding the algorithms, the most important observation is that all evaluations of stencils are independent from each other and could in principle be done in parallel. For example the extreme situation of one processor per grid point would work with as many parallel I/Os as there are points in the stencil, plus one I/O for writing the result, assuming that the model allows multiple reads and writes to the same block of external memory. This algorithm is obviously optimal, but it has many more non-compulsory I/Os than the algorithms we focus on. For moderate P we use the proposed serial algorithms and merely split the computation into P contiguous parts. The only additional non-compulsory I/Os are used to initially fill the local memory. Assuming $P \leq \frac{1}{M} \prod_{i=1}^{n-1} k_i$, this is a lower order term, namely the one that we analyze as the difference between the torus and the grid.

1.3 Related Work

In [1] the external memory or I/O model was introduced by Hong and Kung for $B = 1$. Using essentially an isoperimetric argument they apply it to problems like the Fast-Fourier-Transform (FFT), matrix-matrix multiplication and products of graphs. The latter yields the first bounds for the non-compulsory I/Os of the 1-star stencil $\Theta\left(\frac{1}{n-\sqrt[n]{M}} \cdot \prod_{i=1}^n k_i\right)$. Aggarwal and Vitter generalized Hong and Kungs model in [2] to arbitrary B . Using a counting argument they derive lower bounds for the problems of calculating the FFT, permuting, sorting and matrix transposition. In [4] Savage simplifies Hong and Kungs technique to the S -span approach and in [5] Bezrukov surveys edge isoperimetric problems and shows how they can be used to determine the communication complexity in networks.

The I/O complexity of the 1-star stencil has been discussed further independently by Frumkin and Wijngaart [6,7] and Leopold [8,9,10,11]. The different results for the leading term of the non-compulsory I/Os are given in Table 1 and have to be multiplied by the number of vertices $\prod_{i=1}^n k_i$. Frumkin and Wijngaart consider arbitrary dimensions but focus on the asymptotic behavior of the non-compulsory I/Os. The lower bound uses an isoperimetric argument similar to the one presented in this article but does not exploit its full strength. The best

Table 1. Comparison of the bounds for the leading term of the non-compulsory I/Os for the 1-star stencil. All to be multiplied with the number of grid points $\prod_{i=1}^n k_i$.

	Presented Result	Frumkin and Wijngaart	Leopold
Lower Bound 2D	$\frac{4}{BM}$	$\frac{8}{9} \frac{1}{BM}$	$\frac{2}{BM}$
Lower Bound 3D	$\frac{8}{\sqrt{3}} \frac{1}{B\sqrt{M}}$	$\frac{2}{\sqrt{3}} \frac{1}{B\sqrt{M}}$	$\frac{2}{B\sqrt{M}}$
Low. Bnd. Arbitrary D	$\frac{4 \cdot 2^{1/(n-1)} \cdot (n-1)}{n - \sqrt[n]{n!}} \frac{1}{B^{n - \sqrt[n]{M}}}$	$\left(\frac{2}{3}\right)^{\frac{n}{n-1}} \frac{n}{n - \sqrt[n]{(n-1)!}} \frac{1}{B^{n - \sqrt[n]{M}}}$	n.a.
Upper Bound 2D	$\frac{4}{BM}$	$\mathcal{O}\left(\frac{1}{M}\right)$	$\frac{8}{BM}$
Upper Bound 3D	$\frac{8\sqrt{2}}{\sqrt{3}} \frac{1}{B\sqrt{M}}$	$\mathcal{O}\left(\frac{1}{\sqrt{M}}\right)$	$\frac{4\sqrt{6}}{\sqrt{B}\sqrt{M}}$
Upp. Bnd. Arbitrary D	$4 \cdot 2^{\frac{1}{n-1}} (n-1) \frac{1}{B^{n - \sqrt[n]{M}}}$	$\mathcal{O}\left(\frac{1}{n - \sqrt[n]{M}}\right)$	n.a.

lower bounds are given in [6]. We improve these results by a factor between 4 and 6. The upper bound focuses on the asymptotic behavior and is an existence results. The best upper bound is stated in [7]. Leopold focuses on the two and three dimensional cases. Her lower bounds exploit a weak isoperimetric result [8,10] which we improve by a factor of 2 and $\frac{4}{\sqrt{3}}$ for two respective three dimensions. The upper bounds discuss row and column layouts. By using a data layout suited for our algorithms we decrease the the upper bounds by $\frac{1}{2}$ and $\frac{2}{3\sqrt{B}}$ for two and three dimensions. Leopold also discusses two spatial and one temporal dimension in [9], which is out of the scope of this paper.

There is vast ongoing research about optimizing stencil computations, mostly in two and three dimensions, on modern computer architectures. Since the gap between peak floating point performance and memory bandwidth is growing since years or even decades, this research focuses on improving the I/O behavior of the algorithms. When implementing one may need to be careful about the tradeoff between a more complicated data layout making a sophisticated padding scheme necessary and (theoretically) optimal I/O behavior. Addressing these problems is out of the scope of this work. However, diagonal hyperspace cuts, similar to the ones proven optimal in this work, are often employed in empirical work to select suitable substructures for computation. The literature includes work on compiler optimization [12,13], data layouts determined by space filling curves [14] and wavefront optimization [15]. In the cache oblivious model asymptotic upper bounds are derived in [16] which are then shown to be achieved in [17,18,19]. A recent survey of the field is [20].

2 The Lower Bounds

The lower bound is derived by splitting an arbitrary algorithm into rounds of an equal number of non-compulsory I/Os. The work which can be done in each of these rounds is then bounded by an isoperimetric inequality. This yields the

minimum number of rounds which have to be performed by any algorithm. Multiplying this with the number of non-compulsory I/Os that define a round yields the lower bound. The lower bound is first deduced assuming that an I/O operation accesses one element ($B = 1$) and is then generalized for arbitrary B .

2.1 The Isoperimetric Inequality

The isoperimetric inequality states how many vertices can be enclosed by a fixed number of boundary vertices. The optimal sets in this sense are called isoperimetric sets and as proven by Bollobás and Leader in [21] the isoperimetric sets in \mathbb{Z}_k^n are (fractional) ℓ^1 balls.¹ To state this result precisely we introduce some notation, mainly from [21]:

A fractional system or simply system is a function from \mathbb{Z}_k^n or \mathbb{Z}^n to the unit interval $[0, 1]$. For $f : \mathbb{Z}^n \rightarrow [0, 1]$ the function can take non-zero values only for a finite number of grid points. The weight w of a system f is $w(f) = \sum_{x \in \mathbb{Z}_k^n} f(x)$ or $w(f) = \sum_{x \in \mathbb{Z}^n} f(x)$ according to the domain of f . A fractional system f on \mathbb{Z}_k^n or \mathbb{Z}^n is therefore a generalization of a subset S of \mathbb{Z}_k^n or \mathbb{Z}^n respectively. If a fractional system f takes just the values 0 and 1, then f is naturally identified with the set $S = f^{-1}(1)$ and the weight $w(f)$ is the cardinality of S . The closure ∂f of a system f is given by

$$\partial f(x) = \begin{cases} 1, & \text{if } f(x) > 0 \\ \max\{f(y) : d(x, y) = 1\}, & \text{if } f(x) = 0 \end{cases}.$$

The fractional set

$$b_y^{(r, \alpha)} := \begin{cases} 1, & \text{if } \|x - y\|_1 \leq r \\ \alpha, & \text{if } \|x - y\|_1 = r + 1 \\ 0, & \text{if } \|x - y\|_1 > r + 1 \end{cases}$$

is called the fractional ℓ^1 ball $b_y^{(r, \alpha)}$ of radius $r \in \mathbb{N}_0$, $0 \leq r \leq \frac{k}{2}$, surplus $\alpha \in (0, 1)$ and center $y \in \mathbb{Z}_k^n$. For $0 \leq v \leq k^n$ we also use the notation b_y^v which describes the unique ball of weight v and center y . The following theorem, which states that balls have the smallest closure of all systems of the same weight, was proven by Bollobás and Leader:

Theorem 1 ([21] - Theorem 4: An isoperimetric inequality on the discrete torus). *Let $k \geq 2$ and even, let f be a fractional system on \mathbb{Z}_k^n . Then $w(\partial f) \geq w(\partial b^{w(f)})$.*

We need a version of this theorem which allows us to bound the number of interior vertices given the number of boundary vertices. This differs in two aspects from the above theorem: First, we want to look at the inner-boundary of a set and not at its closure and second we need to have a result for systems of all

¹ It is well known that the isoperimetric sets in the continuous domains \mathbb{R}^n are ℓ^2 balls.

weights but bounded inner-boundary. This will make it necessary to translate Theorem 1 to the infinite grid \mathbb{Z}^n where the boundary of the balls is growing strictly monotonic. To derive the desired result, we need more notation.

Similar to the closure of f we define

$$\Delta f(x) = \begin{cases} 0, & \text{if } f(x) < 1 \\ \min\{f(y) : d(x, y) = 1\}, & \text{if } f(x) = 1 \end{cases} ,$$

the inner core of f , and generalize this to the inner- s -core by applying the operator repeatedly, $\Delta_s f(x) = \underbrace{\Delta \dots \Delta}_{s \text{ times}} f(x)$. This is now used to defined the inner- s -

boundary by $\Gamma_s f(x) = f(x) - \Delta_s f(x)$. This allows to establish the first version of the isoperimetric inequality, stating that balls have the largest inner-core of all systems of the same weight.

Lemma 1 (A version of the isoperimetric inequality).

Let $k \geq 2$ be even, let f be a fractional system on \mathbb{Z}_k^n and $s \in \mathbb{N}$. Then

$$w(\Gamma_s f) \geq w(\Gamma_s b^{w(f)})$$

which is by definition equivalent to

$$w(\Delta_s f) \leq w(\Delta_s b^{w(f)}) .$$

To prove the lemma we need an additional fact.

Lemma 2. For a fractional system f on \mathbb{Z}_k^n the following inequality holds:

$$\partial(\Delta(f(x))) \leq f(x) .$$

Proof. This is seen by examining the different cases carefully. If $f(x) = 0$ it follows that $\partial(\Delta f)(x) = 0$ as well since all neighbors of x are set to 0 by the Δ -operator. When $0 < f(x) < 1$, $\Delta f(x) = 0$ and for all y such that $\|x - y\| = 1$ we have $\Delta f(y) \leq f(x)$ and hence $\partial(\Delta f)(x) \leq f(x)$. When $f(x) = 1$ the claim holds trivially. \square

Proof (of Lemma 1). The claim is proven by induction over s . First, consider the case $s = 1$. If $w(f) \leq 1$ then $w(\Gamma f) = w(f)$ and $w(\Delta f) = 0$ such that the claim holds. Assume there exists some fractional system f with $w(f) > 1$ such that

$$w(\Gamma f) < w(\Gamma b^{w(f)}) \quad \text{and hence} \quad w(\Delta f) > w(\Delta b^{w(f)}) .$$

By the latter and the strict monotonicity of $w(\partial b^{(\cdot)})$ we get

$$w(\partial b^{w(\Delta f)}) > w(\partial b^{w(\Delta b^{w(f)})}) .$$

To simplify the right hand side we use that the inner core of a ball is itself a ball and hence we can discard building the ball of it.

$$w\left(\partial b^{w(\Delta b^{w(f)})}\right) = w\left(\partial \Delta b^{w(f)}\right) .$$

For a ball with $w(f) > 1$ the closure of the inner core is pointwise equal to the ball itself. Furthermore we employ Lemma 2.

$$w\left(\partial \Delta b^{w(f)}\right) = w(b^{w(f)}) = w(f) \geq w(\partial \Delta f) .$$

Reading this sequence of inequalities altogether yields

$$w\left(\partial b^{w(\Delta f)}\right) > w(\partial \Delta f) ,$$

which contradicts Theorem 1 for $\Delta(f)$ as fractional system and proves the claim for $s = 1$.

Let us now prove the claim for s assuming it holds for $s - 1$. Using the induction assumption for Δf we arrive at

$$w(\Delta_s f) = w(\Delta_{s-1} \Delta f) \leq w\left(\Delta_{s-1} b^{w(\Delta f)}\right) .$$

Noting that b , Δb and $\Delta_{s-1} b$ are pointwise monotonically increasing yields that $w(\Delta_{s-1} b)$ is monotonically increasing. Hence we can apply the result proven for $s = 1$ to yield

$$w\left(\Delta_{s-1} b^{w(\Delta f)}\right) \leq w\left(\Delta_{s-1} b^{w(\Delta b^{w(f)})}\right) .$$

But the inner core of a ball is a ball itself, so we can discard building the ball of it and this simplifies to the required result

$$w\left(\Delta_{s-1} b^{w(\Delta b^{w(f)})}\right) = w\left(\Delta_{s-1} \Delta b^{w(f)}\right) = w\left(\Delta_s b^{w(f)}\right) .$$

□

Since the weight of the inner- s -core of a ball is monotonically increasing with the weight of the ball, this result can be used to deduce the implication

$$(w(f) \leq v) \Rightarrow (w(\Delta_s f) \leq w(\Delta_s b^v)) .$$

Nevertheless, we run into problems when bounding the weight of a ball given that its inner-boundary is bounded. The inner-boundary of balls is only monotonically increasing until $v \approx \frac{k^n}{2}$ and thereafter monotonically decreasing. To overcome this problem, we transfer the results to the infinite grid, where the inner- s -boundary of balls is monotonically increasing with respect to the weight of the ball.

Theorem 2 (The boundary bounds the core on \mathbb{Z}^n).

Let $s \in \mathbb{N}$ and f be a fractional system on \mathbb{Z}^n . For $v \in \mathbb{R}_0^+$ the following holds:

$$(w(\Gamma_{2s} f) \leq w(\Gamma_{2s} b^v)) \Rightarrow (w(\Delta_s f) \leq w(\Delta_s b^v)) . \quad (1)$$

Proof. The proof is split into two parts. $(w(\Gamma_{2s}f) \leq w(\Gamma_{2s}b^v)) \Rightarrow (w(f) \leq v)$ is first proven by contraposition. Hence, we first prove

$$(w(\Gamma_{2s}f) > w(\Gamma_{2s}b^v)) \Leftarrow (w(f) > v) .$$

Since f takes just a finite number of non-zero values, we can find k such that all non-zero values of f are in the grid $\{-k, \dots, k\}^n$ and we can embed f in the torus $\mathbb{Z}_{2(k+1)}^n$ such that no points of f touch where the grid is closed to a torus. Hence, the inner- s -boundary on \mathbb{Z}^n is exactly the same as on $\mathbb{Z}_{2(k+1)}^n$. On this torus Lemma 1 yields $w(\Gamma_s f) \geq w(\Gamma_s b^{w(f)})$. To conclude the first part, note that the weight $w(\Gamma_s b^v)$ is strictly monotonically increasing with respect to v on \mathbb{Z}^n . This yields

$$w(\Gamma_s f) \geq w(\Gamma_s b^{w(f)}) > w(\Gamma_s b^v)$$

and since s was arbitrary also $w(\Gamma_{2s}f) > w(\Gamma_{2s}b^v)$ which establishes the first part.

Employing Lemma 1 again on the torus and noting that $w(\Delta_s b^v)$ is monotonically increasing with respect to v yields $(w(f) \leq v) \Rightarrow (w(\Delta_s f) \leq w(\Delta_s b^v))$ and the proof is complete. \square

2.2 The Size of the ℓ^1 Ball and its Boundary

In this section we derive the asymptotic expansion for the number of vertices of a ball and its inner-boundary in \mathbb{Z}^n with respect to the radius r ,

$$w(b^{(r,0)}) = \frac{2^n}{n!} \cdot r^n + \mathcal{O}(r^{n-1}) \quad \text{and} \quad w(\Gamma b^{(r,0)}) = \frac{2^n}{(n-1)!} \cdot r^{n-1} + \mathcal{O}(r^{n-2}) . \quad (2)$$

As long as the sides of the torus or grid are big enough, $k \geq 2(r+1)$, the formulas apply there also. Note that all lower order terms have positive coefficients. We derive these formulas by recursing over the dimensions. Hence it is useful to introduce the notation $b_n^{(r,0)}$ for the ball of radius r in n dimensions. The ℓ^1 ball of dimension n consists of smaller balls of one dimension less, namely the level sets in the new dimension:

$$b_n^{(r,0)} = b_{n-1}^{(r,0)} + 2 \cdot \sum_{l=0}^{r-1} b_{n-1}^{(l,0)} . \quad (3)$$

Another simple fact is $b_n^{(r,0)} = b_n^{(r-1,0)} + \Gamma b_n^{(r,0)}$ which yields when combined with (3)

$$\Gamma b_n^{(r,0)} = b_{n-1}^{(r,0)} + b_{n-1}^{(r-1,0)} . \quad (4)$$

Since $w(\Gamma b_n^{(0,0)}) = 1$ for all $n \in \mathbb{N}$ and $w(\Gamma b_1^{(r,0)}) = 2$ for $r \geq 1$ the weight of the one dimensional balls is given by

$$b_1^{(r,0)} = 2r + 1 . \quad (5)$$

Recursion (3) yields that $w(b_n^{(r,0)})$ and $w(\Gamma b_n^{(r,0)})$ are polynomials in r of degree n and $n-1$ with non-negative coefficients. So they can be written as

$$w(b_n^{(r,0)}) = \sum_{i=0}^n \alpha_{n,i} \cdot r^i \quad \text{and} \quad w(\Gamma b_n^{(r,0)}) = \sum_{i=0}^{n-1} \beta_{n,i} \cdot r^i.$$

Examining the leading term $\alpha_{n,n}$ of $w(b_n^{(r,0)})$ yields

$$\begin{aligned} w(b_n^{(r,0)}) &= w(b_{n-1}^{(r,0)}) + 2 \sum_{l=0}^{r-1} w(b_{n-1}^{(l,0)}) = \mathcal{O}(r^{n-1}) + 2 \sum_{l=0}^{r-1} \sum_{i=0}^{n-1} \alpha_{n-1,i} \cdot l^i = \\ &= \mathcal{O}(r^{n-1}) + 2 \sum_{i=0}^{n-1} \left(\alpha_{n-1,i} \sum_{l=0}^{r-1} l^i \right) \leq \\ &\leq \mathcal{O}(r^{n-1}) + 2 \sum_{i=0}^{n-1} \left(\alpha_{n-1,i} \int_0^r l^i dl \right) = \\ &= \mathcal{O}(r^{n-1}) + 2 \sum_{i=0}^{n-1} \left(\alpha_{n-1,i} \frac{r^{i+1}}{i+1} \right) = 2\alpha_{n-1,n-1} \frac{r^n}{n} + \mathcal{O}(r^{n-1}). \end{aligned}$$

Comparing the coefficient of the leading terms yields the recursion

$$\alpha_{n,n} = \frac{2}{n} \alpha_{n-1,n-1}.$$

The recursion stops with (5), namely $\alpha_{1,1} = 2$. Hence we get

$$\alpha_{n,n} = \frac{2^n}{n!} \quad \text{and} \quad w(b_n^{(r,0)}) = \frac{2^n}{n!} \cdot r^n + \mathcal{O}(r^{n-1}).$$

Now (4) yields

$$\beta_{n,n-1} = \frac{2^n}{(n-1)!} \quad \text{and hence} \quad w(\Gamma b_n^{(r,0)}) = \frac{2^n}{(n-1)!} \cdot r^{n-1} + \mathcal{O}(r^{n-2}).$$

2.3 Pathwidth

We introduce and employ pathwidth [22] to ensure that we are working on the “inside” of the torus and can treat it like the infinite grid which allows to use the isoperimetric results. Furthermore, the pathwidth of the grid ensures that non-compulsory I/Os are performed when the s -star stencil is evaluated. This allows to split an arbitrary algorithm into rounds of a certain number of non-compulsory I/Os.

Definition 1 (Pathwidth [22]). A path decomposition of a graph $G = (V, E)$ is a sequence of subsets of vertices (X_1, X_2, \dots, X_r) , called bags, such that

1. $\bigcup_{1 \leq i \leq r} X_i = V$.

2. for all edges $(v, w) \in E$ there exists an $i \in \{1, \dots, r\}$ such that $x \in X_i$ and $v \in X_i$.
3. for all i, j, k such that $1 \leq i \leq j \leq k \leq r$ it holds that $X_i \cap X_k \subseteq X_j$.

The width of a path decomposition (X_1, X_2, \dots, X_r) is $\max_{1 \leq i \leq r} |X_i| - 1$. The width of a graph G is the minimum width over all possible path decompositions of G .

Condition (3) implies that a vertex can only be in a consecutive block of bags and not reappear after it has been removed from a bag once.

An algorithm evaluating the s -star stencil on G without non-compulsory I/Os and internal memory of size M implies that $\text{pathwidth}(G) \leq M - 1$. If we can evaluate the s -star stencil on G with internal memory of size M and without loading a vertex twice this immediately gives a path decomposition with bags of size at most M . The bags are the different sets of elements the internal memory is containing at different stages of the algorithm. Since the two dimensional grid $[k_1] \times [k_2]$ has pathwidth $\min\{k_1, k_2\}$ (Corollary 89 of [22]) there have to be non-compulsory I/Os if we want to evaluate the s -star stencil on a two dimensional grid or torus with $\min\{k_1, k_2\} \geq M$.

Pathwidth can also be modeled by a *robber and cop game* [23]. This game is played on an arbitrary undirected graph like the grid or the torus. Initially p cops are placed on the vertices of the graph and afterwards the robber chooses its initial position. The robber is visible to the cops during the game and the game proceeds in rounds. First the cops announce where they want to be placed in the next round. Then every cop that wants to move boards a helicopter. While the cops are moving in the air the robber is allowed to move to an arbitrary vertex of the graph if he can reach it without running into a cop. Thereafter the cops land and the robber escapes in that round if no cop lands on the vertex the robber is standing on. The game then continues with the next round. If there is a strategy so that the robber is able to escape the cops for an infinite number of rounds we say that the robber wins.

The following implication is proven in [23]: When the robber cop game is played with p cops on a graph G and if there is a strategy so that the robber wins, G has to have pathwidth bigger than $p - 1$.

Lemma 3. *If the subgraph H of a two dimensional grid or torus consists of $p + 1$ complete rows and complete columns, then $\text{pathwidth}(H) \geq p$.*

Proof. We give a strategy in the robber and cop game such that the robber wins against p cops to prove the claim. Since there are $p + 1$ complete rows and columns in H , the robber is free to start in a row which is empty, after the initial placements of the cops. When the cops announce their move, there will be a free column in the next configuration. Since the robber is in a free row, it can move to this column which is free in the next configuration. The game now proceeds with rows and columns interchanged. The robber always escapes from a free row to a free column and vice versa. \square

2.4 Splitting an Arbitrary Algorithm into Rounds

To derive the lower bounds assume an arbitrary algorithm evaluating the s -star stencil on $\mathbb{Z}_{k_1} \times \dots \times \mathbb{Z}_{k_n}$ is given. When $\min\{k_1, k_2\} \geq M$ the pathwidth of $\mathbb{Z}_{k_1} \times \dots \times \mathbb{Z}_{k_n}$ is at least M and hence the algorithm causes non-compulsory I/O operations. We can count these operations and split the algorithm into rounds of c non-compulsory I/Os. c is denoted round length and hence all rounds except the last one cause c non-compulsory I/Os. The isoperimetric inequality allows to bound the work per round which yields a lower bound for the number of rounds. Multiplying the number of rounds with the number of non-compulsory I/Os c per round establishes the lower bound. This approach is similar to the idea presented by Hong and Kung in [1] and therefore we call the rounds Hong-Kung rounds.

To apply the isoperimetric inequality we need to establish a link between the inner-core and inner-boundary and the rounds. Assume an arbitrary algorithm is split into its Hong-Kung rounds of length c and one round is picked. Denote with S the set of vertices which are in internal memory at some point of this round. Let $Transfer(S)$ be the *transfer vertices* of S . These are all vertices of S which are also available in other rounds. $Eval(S)$, the *evaluated vertices*, are all vertices of S for which the s -point stencil is evaluated in the current round. The following two observations relate these sets to the inner-core and the inner-boundary.

$$\Gamma_{2s}(S) \subset Transfer(S) \quad \text{and} \quad Eval(S) \subset \Delta_s(S). \quad (6)$$

A vertex can only be evaluated in round S if all its neighbors within distance s are in S as well. $\Delta_s(S)$ consists of exactly these vertices. Equivalently $\Gamma_s(S)$ are the vertices which cannot be evaluated in round S . Take any $x \in \Gamma_s(S)$. All vertices which are within distance s from x need to be in the round in which x is evaluated. Hence they need to be transferred. The set of all vertices of S within distance s from any of the vertices of $\Gamma_s(S)$ is $\Gamma_{2s}(S)$. Therefore these vertices are a subset of the transfer vertices.

Furthermore, we can give an upper bound for the transfer vertices per round and hence for the inner- $2s$ -boundary vertices. At the beginning of a round at most M vertices are in internal memory. These vertices have also been in internal memory at the end of the previous round and hence they are transfer vertices. The M vertices which are in internal memory at the end of the round are available to the next round so they are transferred as well. During one round an unlimited number of compulsory I/Os but only c non-compulsory I/Os occur. Assume a vertex does not cause a non-compulsory I/O and, since this case has already been taken into consideration, is not in internal memory at the beginning or end of that round. This vertex has then not been loaded before and hence it has not taken part in any of the previous rounds. Furthermore, since it does not cause a non-compulsory I/O it is also not written back to external memory and hence not available to any succeeding round. In conclusion, at most c additional vertices, associated with the non-compulsory I/Os, are transfer vertices. Altogether, there are at most $2M + c$ transfer vertices per round,

$$w(Transfer(S)) \leq 2M + c. \quad (7)$$

Since all vertices of $S \setminus \text{Transfer}(S)$ are not available for other rounds their s -star stencil has to be computed in the current round. Furthermore, since the vertices of $S \setminus \text{Transfer}(S)$ do not cause non-compulsory I/Os they are loaded into internal memory only during the current round. Hence their pathwidth has to be less or equal to $M - 1$,

$$\text{pathwidth}(S \setminus \text{Transfer}(S)) \leq M - 1. \quad (8)$$

2.5 Deducing the Lower Bound

This section addresses the following issues: An arbitrary algorithm evaluating the s -star stencil on the $\mathbb{Z}_{k_1} \times \dots \times \mathbb{Z}_{k_n}$ is split into its Hong-Kung rounds of c non-compulsory I/Os. Limited pathwidth of $S \setminus \text{Transfer}(S)$ ensures that we can handle the torus like an infinite grid. This allows to apply the isoperimetric inequality to upper bound the vertices which can be evaluated per round and hence lower bound the number of rounds that are performed. This yields a lower bound for the number of non-compulsory I/Os on the torus which we finally transfer to the grid.

So, assume an arbitrary algorithm evaluating the s -star stencil on $\mathbb{Z}_{k_1} \times \dots \times \mathbb{Z}_{k_n}$ is given. Assuming $k_1, k_2 \geq M$ it has to perform non-compulsory I/Os and hence we can split the algorithm into Hong-Kung rounds of c non-compulsory I/Os each. The last round may have less non-compulsory I/Os. Fix one of the rounds and denote the vertices which are in internal memory in this round by S . We know $w(\text{Transfer}(S)) \leq 2M + c$ and $\text{pathwidth}(S \setminus \text{Transfer}(S)) \leq M - 1$.

Assuming $k_1, k_2 \geq 2M + c + (M + 1)$ and $k_i \geq 2M + c + 1$ for $i \in \{3, \dots, n\}$ the vertices of (at least) $M + 1$ hyperplanes of normal x_1 , $M + 1$ hyperplanes of normal x_2 and one hyperplane of normal x_i ($3 \leq i \leq n$) do not belong to $\text{Transfer}(S)$. These hyperplanes form a connected component in $\mathbb{Z}_{k_1} \times \dots \times \mathbb{Z}_{k_n}$. So they could either be a subset of $S \setminus \text{Transfer}(S)$ or disjoint from S . Taking the intersection of all hyperplanes results in a subset of a two dimensional torus of at least $M + 1$ complete rows and columns. Hence, by Lemma 3 we have $\text{pathwidth}(S \setminus \text{Transfer}(S)) \geq M$ which contradicts (8). Therefore, there is at least one hyperplane for each normal direction x_i ($1 \leq i \leq n$) which is disjoint from S . Deleting these hyperplanes results in a graph which can be embedded in the infinite grid \mathbb{Z}^n .

Combining (6) and (7) we get $w(\Gamma_{2s}S) \leq 2M + c$. Denote with v_0 the weight such that

$$w(\Gamma_{2s}b^{v_0}) = 2M + c. \quad (9)$$

By Theorem 2 and (6) the number of evaluable vertices of S is bounded by $w(\text{Eval}(S)) \leq w(\Delta_s b^{v_0})$. This establishes the framework for the lower bound, namely the lower bound is given by

$$\frac{c}{w(\Delta_s b^{v_0})} \cdot \prod_{i=1}^n k_i. \quad (10)$$

It remains to determine v_0 which in turn depends upon c . Since s is assumed to be small and constant we simplify (9) before solving. Denote (r_0, α_0) the radius and surplus such that $b^{v_0} = b^{(r_0, \alpha_0)}$. Using (2) the asymptotics of $w(\Gamma_{2s}b^{v_0})$ is given by

$$\begin{aligned}
w(\Gamma_{2s}b^{v_0}) &= w\left(\Gamma b^{(r_0, \alpha_0)}\right) + w\left(\Gamma_{2s-1}b^{(r_0, 0)}\right) + w\left(\Gamma b^{(r_0-(2s-1), (1-\alpha_0))}\right) = \\
&= \alpha_0 \cdot \frac{2^n}{(n-1)!}(r_0+1)^{n-1} + \mathcal{O}(r_0^{n-2}) + \sum_{k=0}^{2s-2} \frac{2^n}{(n-1)!}(r_0-k)^{n-1} + \mathcal{O}(r_0^{n-2}) + \\
&\quad + (1-\alpha_0) \cdot \frac{2^n}{(n-1)!}(r_0-(2s-1))^{n-1} + \mathcal{O}(r_0^{n-2}) = \\
&= 2s \cdot \frac{2^n}{(n-1)!}r_0^{n-1} + \mathcal{O}(r_0^{n-2}) .
\end{aligned} \tag{11}$$

Since all coefficients in the lower order terms are non-negative, dropping the lower order terms before solving (9) increases r_0 and v_0 , increases $w(\Delta_s b^{v_0})$ and hence weakens the lower bound. Solving (11) without the lower order terms yields

$$r_0 = \sqrt[n-1]{(n-1)! \cdot \frac{2M+c}{2s \cdot 2^n}} . \tag{12}$$

Plugging that into (10) for the lower bound and maximizing over c (setting the derivative to 0 and checking that the solution is a maximum) yields that the best round length, disregarding lower order terms, is approximately

$$c = 2(n-1) \cdot M.$$

Using this round length, we determine an upper bound r_0 for the radius of a ball to be handled in one round using (12)

$$r_0 = \sqrt[n-1]{\frac{n!}{2^n} \frac{M}{s}}$$

and finally, by plugging this radius into (10), the lower bound

$$\begin{aligned}
\left(\prod_{i=1}^n k_i\right) \cdot \frac{2(n-1)M}{w(\Delta_s b^{(r_0, 0)})} &= \frac{2(n-1)M}{w(b^{((r_0-s), 0)})} \cdot \prod_{i=1}^n k_i = \frac{2(n-1)M}{\frac{2^n}{n!} \cdot r_0^n + \mathcal{O}(r_0^{n-1})} \cdot \prod_{i=1}^n k_i = \\
&= \frac{2(n-1)M}{\left(\frac{n!}{2^n} s^n\right)^{1/(n-1)} M^{n/(n-1)} + \mathcal{O}(M)} \cdot \prod_{i=1}^n k_i = \\
&= \frac{2(n-1)}{\left(\frac{n!}{2^n} s^n\right)^{1/(n-1)} M^{1/(n-1)} + \mathcal{O}(1)} \cdot \prod_{i=1}^n k_i = \\
&= \left(\frac{2(n-1)M}{\left(\frac{n!}{2^n} s^n\right)^{1/(n-1)} M^{1/(n-1)}} - \mathcal{O}\left(\frac{1}{n^{-1/\sqrt{M^2}}}\right) \right) \cdot \prod_{i=1}^n k_i = \\
&= \left(4(n-1) \sqrt[n-1]{\frac{2}{n!} s^n} \cdot \frac{1}{n^{-1/\sqrt{M}}} - \mathcal{O}\left(\frac{1}{n^{-1/\sqrt{M^2}}}\right) \right) \cdot \prod_{i=1}^n k_i .
\end{aligned}$$

This bound was derived on the torus $\mathbb{Z}_{k_1} \times \cdots \times \mathbb{Z}_{k_n}$ and we can apply it to the grid $[k_1] \times \cdots \times [k_n]$ using a reduction. Let $b_n^{(r, \alpha)}$ denote the ℓ^1 ball of radius r and surplus α in n dimensions.

Lemma 4. *Any algorithm using internal memory of size M and evaluating the s -point stencil on the grid $[k_1] \times \cdots \times [k_n]$ induces an algorithm, using the internal memory M and evaluating the s -point stencil, on the torus $\mathbb{Z}_{k_1} \times \cdots \times \mathbb{Z}_{k_n}$ causing at most $\mathcal{O}\left(\prod_{i=1}^{n-1} k_i\right)$ additional I/Os.*

Proof. When the algorithm for the grid is evaluated on the torus, only the vertices close the boundary of the grid have to be treated differently. If a vertex is within ℓ^1 distance $s-1$ in a unit direction from a bounding hyperplane, at most half of the points of the s -point stencil, corresponding to that unit direction, have to be read and written additionally for this vertex. Altogether these are at most $2n \cdot s \cdot \frac{b_n^{(s,0)}}{2} \cdot \prod_{i=1}^{n-1} k_i = \mathcal{O}\left(\prod_{i=1}^{n-1} k_i\right)$ I/Os. \square

Hence the lower bound on the torus and grid of the same side lengths differ at most by an additive constant of $\mathcal{O}\left(\prod_{i=1}^{n-1} k_i\right)$. For arbitrary B one I/O operation affects at most B elements, such that the total number of I/Os (including compulsory ones) for the grid is

$$C(k_1, \dots, k_n) = \left(2 + \frac{4(n-1) \cdot {}^{n-1}\sqrt{\frac{2}{n!} s^n}}{{}^{n-1}\sqrt{M}} - \mathcal{O}\left(\frac{1}{{}^{n-1}\sqrt{M^2}} + \frac{1}{k_n}\right)\right) \frac{\prod_{i=1}^n k_i}{B}.$$

3 The Upper Bounds

The algorithms evaluating the s -point stencil depend on the layout in which the grid data is saved. In two dimensions, for example, the lower bound suggests to work in adjacent ℓ^1 balls where neighboring balls form a diagonal working band. Although, for $B = 1$ or a data layout supporting this data access, the diagonal sweep through the data provides a matching upper bound, it is not optimal for many other data layouts.

All the bounds have in common that a *sweep shape* is moved through the grid in unit shifts in a *sweep order* resulting in *working bands*. The sweep shapes are line segments in two dimensions and subsets of planes in three dimensions. To achieve good results, the data layout has to reproduce the shape of the sweep shapes, and to evaluate all vertices of the grid, the working bands have to overlap dividing the working band into *core* and *wing bands*. For optimal asymptotic behavior vertices in different bands have to be saved in separate blocks.

The description of the data layouts is given indirectly by the core and wing bands of the algorithm. For each subset of working bands that have overlapping wing bands, the data is save in separate blocks. Within the bands the data is ordered in the natural way the algorithm accesses it. Instead of already having

the data separated by core and wing bands the algorithm can change the data layout while it is executed.

Having the preceding s and the proceeding s sweep shapes in internal memory allows to evaluate the s -star stencils of the current sweep shape. Special attention has to be paid to evaluate the vertices which are in the relative boundary of the sweep shape. The relative boundary is the boundary of the sweep shape within its line or plane and these vertices are evaluated by overlapping the working bands. In fact only $2s$ instead of $2s + 1$ full sweep shapes have to be kept in internal memory, as the vertices of the s .th preceding sweep shape can be deleted or written to the external memory as vertices of the s .th proceeding sweep shape are loaded. Only a constant number of vertices, depending on s but independent of the size of the sweep shape, are necessary in addition to $2s$ layers of sweep shapes.

When discussing the upper bounds we focus on the leading term of the non-compulsory I/Os and disregard the constant number of compulsory I/Os. The exact analysis of the lower order terms is tedious work and presented separately in Section 4. The detailed analysis keeps repeating the same steps for every algorithm and is mainly bookkeeping of the different lower order terms.

3.1 Upper Bounds in 2 Dimensions

The upper bounds in two dimensions are summarized in Table 2. When the sweep shape is shifted orthogonal to the alignment of the blocks (block aligned column layout) the I/O behavior is asymptotically optimal. However, the constant at the leading term of the non-compulsory I/Os matches the lower bound only for the block aligned diagonal layout.

The main results for two dimensional row and column layouts have been presented by Leopold in [8]. In a row (column) layout blocks extend in x_1 -direction (x_2 -direction). Sweeping through a row layout in x_1 -direction (x_2 -direction) is equivalent to sweeping in a column layout in x_2 -direction (x_1 -direction). Hence only sweeps in x_1 -direction will be discussed.

Table 2. Upper bounds for different data layouts in two dimensions

Layout	Sweep Shape	Sweep Order	Upper Bound
Row	Vertical	x_1	$\frac{8s}{M} \cdot k_1 k_2$
Block Aligned Column	Vertical	x_1	$\frac{8s^2}{BM} \cdot k_1 k_2$
Block Aligned Diagonal	Diagonal	x_1, x_2	$\frac{4s^2}{BM} \cdot k_1 k_2$

Row Layout: The sweep shape is a vertical line segment of m points shifted in x_1 direction. The middle $m - 2s$ vertices of a sweep shape can be evaluated

when $2s$ sweep shapes are in internal memory. For $B \geq 2s$ one block has to be in internal memory per row of the sweep shape ($mB \leq M$). In the worst case of a block aligned layout the $2s$ previous sweep shapes are also needed in internal memory which adds an error term of $\mathcal{O}\left(\frac{k_1 k_2}{BM}\right)$ (see (17) in Sect. 4). Disregarding this, the size of the sweep shape is roughly $m = \frac{M}{2B}$. The s uppermost and s lowermost rows of a working band cannot be evaluated which adds an error term of $\mathcal{O}\left(\frac{k_1 k_2 B}{M^2}\right)$ (see (15), (16)). This error term also accounts for reserving blocks for the output. Assuming the whole working band can be evaluated, there are about $\frac{k_2}{m}$ working bands. The $2s$ uppermost and $2s$ lowermost rows of a working band overlap with the according rows of neighboring working bands and hence they cause non-compulsory I/Os. When evaluating the working bands bottom up, the lowermost (uppermost) rows cause non-compulsory reads (writes). Each B columns there is one such I/O. Altogether, the number of non-compulsory I/Os is roughly $\frac{k_2}{m} \frac{k_1}{B} \cdot 2s \cdot 2 = \frac{4s}{M} \cdot k_1 k_2$. There is an additional error term of $\mathcal{O}(k_2)$ (see (14)) since the first and last block of each row, including the core bands, may be loaded twice. This bound is a factor of B worse than the best upper bound since, per row of a working band, a whole block instead of only $2s$ vertices reside in internal memory. Please see Sect. 4 for a detailed analysis of the lower order terms and an explanation what they capture.

Block Aligned Column Layout: In a block aligned column layout the blocks extend in x_2 -direction and start at the same x_2 values for all columns (see Fig. 1 - middle). The sweep shape is a vertical line segment of m vertices. To evaluate a sweep shape, the s preceding and s proceeding sweep shapes and an additional $s + 1$ vertices of the s .th preceding sweep shape have to be in internal memory. So we have roughly $m = \frac{M}{2s}$ and hence $\frac{k_2}{m}$ working bands.

When the vertices of core and wing bands are not saved in separate blocks, but assuming $B \geq 2s$ and the working band size is chosen such that the wing bands are totally within one block (otherwise multiply the non-compulsory I/Os by 2) one I/O occurs for every column at the top and bottom end of the working band. Hence the non-compulsory I/Os are $\frac{k_2}{m} \cdot k_1 \cdot 2 = \frac{4s}{M} k_1 k_2$. Since there are two I/Os every column, each writing a full block instead of the necessary $2s$ elements, this bound is a factor of B of the best upper bound.

Now consider saving the vertices that are part of overlapping working bands, defining the wing bands, in separate blocks. Each wing band contain $2s$ vertices per column. So there are 2 I/Os every $\frac{2s}{B}$ columns and the upper bound is $\frac{k_2}{m} \cdot 2 \cdot \frac{k_1 \cdot 2s}{B} = 8s^2 \frac{k_1 k_2}{MB}$. This analysis estimates the leading term of the non-compulsory I/Os correctly. Please refer to Sect. 4 for the detailed analysis including and explaining the lower order terms. The upper bound, including the compulsory I/Os, is

$$\left(2 + \frac{8s^2}{M} + \mathcal{O}\left(\frac{1}{M^2} + \frac{1}{k_2}\right)\right) \cdot \frac{k_1 k_2}{B}.$$

This layout achieves the correct asymptotics but the leading term of the non-compulsory I/Os is by a factor 2 worse than the lower bound. In [8] Leopold uses

a mixed row/column layout which achieves the same constant at the leading term of the non-compulsory I/Os.

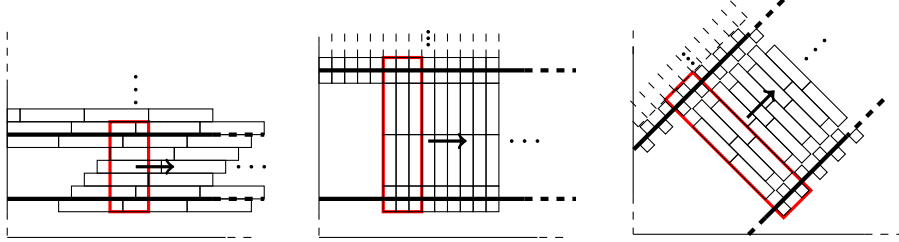


Fig. 1. The row (left), block aligned column (middle) and block aligned diagonal layout (right) in two dimensions for $s = 1$. Current $2s + 1$ sweep shapes red.

Block Aligned Diagonal Layout: The lower bound suggests to work in adjacent ℓ^1 balls. Within a ball we work in diagonals of direction $(1, -1)$ which we shift alternately in both unit directions. These diagonal sweep shapes consist of m vertices. There is a close connection between the sweep shape and the path-width and the minimum bisection separator of the ℓ^1 ball as graph. The internal memory must hold $2s$ sweep shapes, hence $m = \frac{M}{2s}$. The working bands and their overlap again define wing and core bands and the data is saved in separate blocks for each of these bands (see Fig. 1 - right). Within a band the data is ordered by sweep shapes the algorithm accesses successively. Per row, a core band has $2m - 4s$ and a wing band $2s$ vertices. Compared to the column layout the width of a core band has doubled. Hence the upper bound, matched by the lower bound and including the compulsory I/Os and the lower order terms derived in Sect. 4, is

$$\left(2 + \frac{4s^2}{M} + \mathcal{O}\left(\frac{B}{M^2} + \frac{1}{k_1}\right)\right) \cdot \frac{k_1 k_2}{B}.$$

3.2 Upper Bounds in 3 Dimensions

The three dimensional results are presented in Table 3. Working orthogonal to the alignment of the blocks (block aligned column/pole layout) achieves optimal asymptotic behavior. Using an two dimensional ℓ^1 ball instead of a square as sweep shape improves the leading term of the non-compulsory I/Os by a factor of $\frac{1}{\sqrt{2}}$ (2D block aligned diagonal layout). The best new upper bound improves this by another factor of $\frac{1}{\sqrt{3}}$, leaving a gap of $\sqrt{2}$ to the lower bound (hexagonal aligned diagonal layout). The algorithm shifts a hexagonal sweep shape, resulting from the intersection of a three dimensional ℓ^1 ball with a plane of normal $(1, 1, 1)$ through the origin, and alternates the three unit shifts (hexagonal

aligned diagonal layout). Refer to Section 4 for the derivations of the lower order terms.

Table 3. Upper bounds for different data layouts in three dimensions

Layout	Sweep Shape	Sweep Order	Upper Bound
Row	Square	x_1	$\frac{8s}{\sqrt{B}\sqrt{M}} \cdot k_1 k_2 k_3$
Block Aligned Column/Pole	Square	x_1	$\frac{8\sqrt{2}s^{3/2}}{B\sqrt{M}} \cdot k_1 k_2 k_3$
2D Block Aligned Diagonal	ℓ^1 ball	x_1	$\frac{8s^{3/2}}{B\sqrt{M}} \cdot k_1 k_2 k_3$
Hexagonal Aligned Diagonal	Hexagonal	x_1, x_2, x_3	$\frac{8\sqrt{2}s^{3/2}}{\sqrt{3}B\sqrt{M}} \cdot k_1 k_2 k_3$

In three dimensions up to four working bands overlap. Per sweep shape, the number of these overlapping vertices solely depends on s and is independent of the size of the sweep shape. Therefore this effect is captured in the the lower order terms and hence we can simplify the analysis such that every vertex in a wing band causes one non-compulsory write and one non-compulsory read operation.

Poles are the equivalents to rows (x_1 -direction) and columns (x_2 -direction) in x_3 -direction. As in the two dimensional case, we consider only sweeps in x_1 -direction but all layouts.

Row Layout: As sweep shape we utilize a square with m vertices per side. Assuming $B \geq 2s$ one block contains all necessary $2s$ sweep shapes and the size of the sweep shape is about $m = \sqrt{\frac{M}{B}}$. A working band has m vertices in x_3 and also in x_2 direction and is surrounded by $4 \cdot m \cdot 2s$ wing vertices per $x_2 x_3$ -plane which each cause one I/O every B $x_2 x_3$ -planes. Hence, the leading term of the non-compulsory I/Os is $\frac{k_3}{m} \frac{k_2}{m} k_1 \frac{4m \cdot 2s}{B} = \frac{8s}{\sqrt{B}\sqrt{M}} \cdot k_1 k_2 k_3$. A detailed analysis like presented in Sect. 4 yields

$$\left(2 + 8s \cdot \frac{\sqrt{B}}{\sqrt{M}} + \mathcal{O} \left(\frac{\sqrt{B}}{M} + \frac{1}{\sqrt{B}\sqrt{M}} \right) \right) \cdot \frac{k_1 k_2 k_3}{B}. \quad (13)$$

The first error term describes that not the full wing band can be evaluated and hence the estimation of the number of working bands contributes an error term. If the layout is block aligned, m blocks as well as the $2s$ precious sweep shapes need to be in memory which is captured by the second error term.

Block Aligned Column and Pole Layout: When sweeping in x_1 direction the column and the pole layout are the same with the role of the coordinates

interchanged. In both layouts the sweep shape is a square of m vertices per side and the blocks are aligned with the sweep shape. Hence we can choose $m = \sqrt{\frac{M}{2s}}$. A working band has m vertices in x_3 and also in x_2 direction and a working band is surrounded by $4 \cdot m \cdot 2s$ wing vertices per x_2x_3 -plane. Since these vertices are sorted according to x_2x_3 -planes each B vertices there is an I/O. Altogether this yields $\frac{k_3}{m} \frac{k_2}{m} k_1 \frac{4m \cdot 2s}{B} = \frac{8\sqrt{2}s^{3/2}}{B\sqrt{M}} \cdot k_1 k_2 k_3$. This layout outperforms the row layout by a factor of $\frac{\sqrt{2s}}{\sqrt{B}}$ and is asymptotically optimal. Since we assumed $B \geq 2s$ for the row layout, the block aligned column or pole layout is always advantageous.

The dominating error term is $\mathcal{O}\left(\frac{\sqrt{B}}{M}\right) \cdot \frac{k_1 k_2 k_3}{B}$. The error term is due to the layout separating wing and core bands of input and output into different blocks.

2D Block Aligned Diagonal Layout: The two dimensional diagonal layout also sweeps in x_1 -direction but takes advantage of the better relative interior to relative boundary ratio of the two dimensional ℓ^1 ball compared to this value for a square. The sweep shape is a two dimensional ℓ^1 ball which lies in an x_2x_3 -plane. Therefore, the data layout in each x_2x_3 -plane is a two dimensional block aligned diagonal layout.

By (2) an ℓ^1 ball of m vertices per side (radius $m - 1$) consists of about $2m^2$ vertices of which $4m$ are boundary vertices. Hence the size of the sweep shape can be chosen as $m = \frac{1}{2}\sqrt{\frac{M}{s}} \cdot 2s$ boundary layers of the two dimensional ℓ^1 ball, about $8ms$ vertices, are part of the wing bands which are saved separately as the algorithm proceeds. Each B vertices cause one I/O. When the grid is tiled with the working bands the offset of the working bands is m in x_2 -direction but $2m$ in x_3 -direction (see Fig. 2). Hence the upper bound is $\frac{k_3}{2m} \frac{k_2}{m} \cdot k_1 \cdot \frac{8ms}{B} = \frac{8s^{3/2}}{B\sqrt{M}} \cdot k_1 k_2 k_3$. The leading error term $\mathcal{O}\left(\frac{\sqrt{M}}{B}\right) \cdot \frac{k_1 k_2 k_3}{B}$ is due to separating the layout into separate blocks for wing and core bands.

Hexagonal Aligned Diagonal Layout: Imitating the lower bound by tiling the grid with ℓ^1 balls fails in three dimensions. Nevertheless, a diagonal sweep is again advantageous. As sweep shape the intersection of the ℓ^1 ball $b_3^{(2m,0)}(0)$ with a plane through the origin and normal $(1, 1, 1)$ is employed (see Fig. 2). This hexagonal sweep shape consists of $2 \cdot \left(\sum_{k=m+1}^{2m} k\right) + (2m + 1) = 3m^2 + 3m + 1$ vertices. In the same way the diagonal line sweep shape in two dimensions has been obtained and this shape is also closely related to pathwidth and the minimum bisection separator of the ℓ^1 ball. We shift the sweep shape alternately in the three unit directions x_1 , x_2 and x_3 to get the working band. Because of the hexagonal structure of the sweep shape we can tile the grid with the working bands.

Figure 2 shows the intersection of a working band with an x_1x_2 -plane. The intersection of the same working band with a different x_1x_2 -plane is obtained by shifting the intersection by $(1, 1, 1)$. The intersection consists of $3 \cdot (3m^2 + 3m + 1)$

vertices. To determine which vertices of these belong to wing bands regard how the shifts of the sweep shape affect the s -star stencil.

Consider the intersection of a working band with an x_1x_2 and a point x in this intersection in its two dimensional representation. Having the points $P_s(x) = \{y \in [k_1] \times [k_2] : \|x - y\|_1 \leq s\} \cup \{y : \|x - y\|_\infty \leq s \wedge x_i \leq 0 \wedge y_i \leq 0 \ (1 \leq i \leq n)\} \cup \{y : \|x - y\|_\infty \leq s \wedge x_i \geq 0 \wedge y_i \geq 0 \ (1 \leq i \leq n)\}$ in the intersection of a working band with an x_1x_2 -plane allows to evaluate x during the sweep. We call P_s the two dimensional projection of the s -star stencil (see Fig. 2). We apply P_s to the intersection of a working band with a x_1x_2 -plane to determine which of these vertices are part of wing bands. This yields $24ms$ vertices per working band and x_1x_2 -plane.

The data layout saves the data grouped according to core bands and wing bands shared by different working bands. Within the working bands the vertices are ordered in increasing distance to $(1, 1, 1)$, i.e. they are grouped by sweep shapes. Within a sweep shape the vertices are ordered with increasing z - and finally with increasing y -coordinate. Using this data layout just, a constant number of vertices, depending only on s , are needed in internal memory in addition to the $2s$ sweep shapes. Therefore every B vertices an I/O is done. The size of a sweep shape can be chosen as $m = \sqrt{\frac{M}{6s}}$. It is left to determine the number of working bands. Dividing k_1k_2 by the number of vertices of the intersection of a working band with a xy -plane estimates the number of working bands. Incomplete working bands at the boundary of the grid can be disregarded as these contribute only lower order terms. Hence the upper bound is given by

$$k_3 \cdot \frac{k_1k_2}{9m^2} \cdot \frac{24ms}{B} = \frac{k_1k_2k_3}{B} \cdot \frac{8\sqrt{2}s^{3/2}}{\sqrt{3}\sqrt{M}}.$$

See Sect. 4 for the analysis including the lower order terms. The leading error term is $\mathcal{O}\left(\frac{\sqrt{B}}{M}\right) \cdot \frac{k_1k_2k_3}{B}$ and is due to reserving separate blocks for wing and core bands.

3.3 An Upper Bound for Arbitrary Dimensions

This section gives a simple upper bound for arbitrary dimensions using a block aligned column layout. Sweeping through an n dimensional grid with an $n - 1$ dimensional hypercube lying in a hyperplane of normal x_1 and sweeping it in x_1 direction yields an easy (asymptotically matching) upper bound. We assume that the data is ordered according to core bands and wing bands that belong to different intersections of working bands. One side of the hypercube is then bounded by

$$m = \sqrt[n-1]{\frac{M}{2s}}.$$

An $(n - 1)$ dimensional hypercube has $2n$ faces such that $2(n - 1)$ faces of the working band are surrounded by wing bands. The wing bands have thickness $2s$. The intersection of one wing band shared by two working bands with a

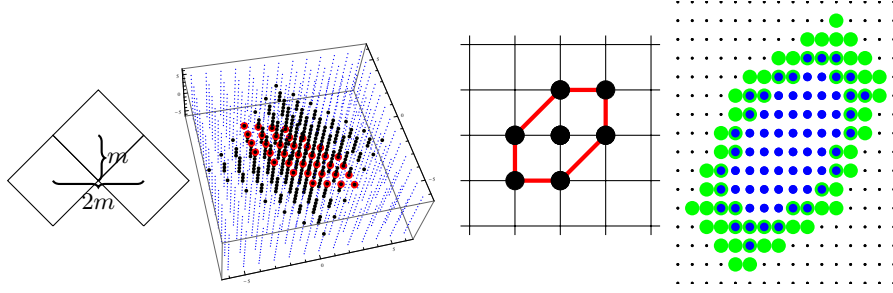


Fig. 2. *Left:* Tiling a 2D grid with ℓ^1 balls; *Middle left:* Intersecting a 3D ℓ^1 ball with plane of normal $(1, 1, 1)$; *Middle right:* Projection of 1-star stencil P_1 ; *Right:* Intersection of a hexagonal working band with an xy-plane. Vertices that can be evaluated blue and wing bands green for $s = 1$.

hyperplane of normal x_n consists of a $n - 2$ dimensional hypercube of side length m . There is a working band every m planes for the first $n - 1$ directions. This yields an upper bound of about

$$\begin{aligned} & \left(\prod_{i=1}^{n-1} \frac{k_i}{m} \right) \cdot \frac{2(n-1) \cdot 2s \cdot m^{n-2} \cdot k_n}{B} = \frac{4s(n-1)}{mB} \prod_{i=1}^n k_i = \\ & = 4 \cdot 2^{\frac{1}{n-1}} \cdot s^{\frac{n}{n-1}} (n-1) \cdot \frac{\prod_{i=1}^n k_i}{B^{\frac{n-1}{n-1}} \sqrt[n]{M}}. \end{aligned}$$

The leading error term is

$$\mathcal{O} \left(\sqrt[n-1]{\frac{B}{M}} \right) \cdot \frac{\prod_{i=1}^n k_i}{B^{\frac{n-1}{n-1}} \sqrt[n]{M}}$$

and rises from reserving whole blocks for every wing band that is shared by different working bands. The quotient of the leading term of the non-compulsory I/Os between this upper bound and the lower bound is $\sqrt[n-1]{n!} \approx \frac{n}{e}$ for large dimensions.

4 Detailed Analysis of the Upper Bounds

The exact analysis of the lower order terms presented in this section is tedious work. The detailed analysis keeps repeating the same steps for every algorithm and is mainly bookkeeping of the different lower order terms.

For the detailed analysis of the upper bounds we assume that the data is given in the layout separated into core bands and wing bands such that no reordering of the data is necessary. Additionally, for each subset of working bands that share wing bands, the data is also saved in separate blocks. The number of working bands that are adjacent to each other is at most 2^n . This and also the number of subsets of overlapping working bands solely depends on n .

We make intense use of the assumptions $\frac{k_i}{M} \rightarrow \infty$ for $i \in \{1, \dots, n\}$ and $\frac{M}{B} \rightarrow \infty$ and regard everything that grows slower than the leading term of the non-compulsory I/Os as lower order terms. In particular terms that solely depend on n or s are regarded constant.

To account for all the I/Os that could occur during the algorithm we first derive a more complicated formula for the upper bound taking into account the difference between working band and the vertices that can be evaluated in a working band, overlap of more than two working bands, blocking the working band into several parts belonging to different intersections of wing bands, reserving blocks for the output, etc. In this first estimation we are sometimes overly pessimistic to simplify the formula for the upper bound. This formula is then simplified step by step while we keep book of all error terms that are dropped. When an error term is calculated, we summarize in parentheses what the error term accounts for.

When simplifying denominator of upper bounds we use the following observation:

$$\frac{1}{m-a} = \frac{1}{m} + \frac{a}{m^2 - ma} .$$

As the output is written in whole blocks we have to save space in internal memory to gather the results of every separately blocked intersection of wing bands. As discussed this number is constant and so, for some constant c , $M - cB$ spots in internal memory are left to load the input data.

4.1 Analysis of the Lower Order Terms of the Two Dimensional Row Layout

This section details the analysis of the two dimensional row layout first discussed in Sect. 3.1. To ensure that only a constant number of output blocks is needed, the data has to be written back in block aligned column layout. Otherwise an output block would be needed for every row of the sweep shape making the number of output blocks dependent on the size of the sweep shape. This change in layout is only needed for the two and three dimensional row layout, as only there vertices that are evaluated consecutively are saved in different blocks. This reordering ensures that we need only $c = 3$ output blocks.

Now consider the input. For $B \geq 2s$ one block has to be in internal memory per row of the sweep shape. In the worst case the layout is block aligned so that the $2s$ previous sweep shapes are also needed. Since we do not need to write the vertices of the core band we reorder them in internal memory so that at most $2s \cdot (m - 4s)$ have to stay in internal memory. The $2s \cdot 4s$ vertices of the wing bands are needed to evaluate other working bands. To preserve the data layout the whole block containing the $2s$ vertices is kept in memory. This gives

$$m \cdot B + \left(\left\lceil \frac{2s \cdot (m - 4s)}{B} \right\rceil + 1 \right) B + 4s \cdot B \leq M - cB .$$

Capturing in $c' = c + 2 + 4s$ the effects of the additional blocks for the wing bands and core bands, this solves to

$$m = \left\lfloor \frac{M - c'B + 8s^2}{B + 2s} \right\rfloor .$$

$\left\lceil \frac{k_2}{m-2s} \right\rceil$ estimates the number of working bands. Per row there are at most $\left\lceil \frac{k_1}{B} \right\rceil + 1$ blocks. Per working band each of these blocks causes an I/O for each of the $4s$ wing rows. Additionally, if the block does not end when the row ends, we need another $2m$ I/Os per working band to write and read the blocks covering two different rows. This yields an upper bound of

$$\left\lceil \frac{k_2}{m-2s} \right\rceil \left(2m + \left(\left\lceil \frac{k_1}{B} \right\rceil + 1 \right) 4s \right)$$

which we now simplify step by step collecting lower order terms. Consider the second term first. Simplifying the second term to $\frac{k_1}{B} \cdot 4s$ yields an error term of at most

$$\left\lceil \frac{k_2}{m-2s} \right\rceil \cdot (2m + 2 \cdot 4s) = \mathcal{O}(k_2) . \quad (14)$$

(If every block spans two rows, there is one additional I/O per row.) Now simplifying the first term to $\frac{k_2}{m-2s}$ adds an error of about

$$1 \cdot \frac{k_1}{B} \cdot 4s = \mathcal{O}\left(\frac{k_1}{B}\right) .$$

(Contribution of the wing rows of the last (partial) working band.) So we are left at

$$\frac{k_2}{m-2s} \cdot \frac{k_1}{B} \cdot 4s .$$

By simplifying the denominator to m , we collect another error term of

$$k_2 \cdot \frac{2s}{m^2 - 2ms} \cdot \frac{k_1}{B} \cdot 4s = \mathcal{O}\left(k_1 k_2 \frac{B}{M^2}\right) \quad (15)$$

(Core band is smaller than working band .) to arrive at

$$\frac{k_2}{m} \cdot \frac{k_1}{B} \cdot 4s = \frac{k_2}{\left\lfloor \frac{M - c'B + 8s^2}{B + 2s} \right\rfloor} \cdot \frac{k_1}{B} \cdot 4s \leq k_2 \frac{B + 2s}{M - c'B + 8s^2 - B - 2s} \cdot \frac{k_1}{B} \cdot 4s .$$

First simplify the denominator to M , collecting

$$k_2 \cdot \frac{k_1}{B} \cdot 4s \cdot (B + 2s) \cdot \frac{c'B - 8s^2 + B + 2s}{M^2 - M c'B + 8s^2 M - MB - 2sM} = \mathcal{O}\left(k_1 k_2 \frac{B}{M^2}\right) . \quad (16)$$

(Output blocks, work in full blocks.) Now simplify the enumerator to B to arrive at

$$k_2 \cdot \frac{B}{M} \cdot \frac{k_1}{B} \cdot 4s = \frac{4s}{M} \cdot k_1 k_2$$

collecting the last error term of

$$k_2 \cdot \frac{k_1}{B} \cdot 4s \cdot \frac{2s}{M} = \mathcal{O}\left(\frac{k_1 k_2}{BM}\right). \quad (17)$$

(m blocks and $2s$ sweep shapes have to in in internal memory.) Let us sum up and simplify the error terms:

$$\mathcal{O}\left(k_2 + \frac{k_1}{B} + k_1 k_2 \frac{B}{M^2} + \frac{k_1 k_2}{BM}\right) = \mathcal{O}\left(k_2 + k_1 k_2 \frac{B}{M^2} + \frac{k_1 k_2}{BM}\right).$$

This yields the final upper bound of

$$\frac{4s}{M} \cdot k_1 k_2 + \mathcal{O}\left(k_2 + k_1 k_2 \frac{B}{M^2} + \frac{k_1 k_2}{BM}\right) = \left(\frac{4s}{M} + \mathcal{O}\left(\frac{1}{k_1} + \frac{B}{M^2} + \frac{1}{BM}\right)\right) \cdot \frac{k_1 k_2}{B}.$$

4.2 Analysis of the Lower Order Terms of the Two Dimensional Block Aligned Column Layout

This section derives the lower order terms that apply to the two dimensional block aligned column layout. $c = 3$ blocks have to reserved for the output of the core band and its two wing bands. Of the core band at most $2s$ sweep shapes and s vertices have to be in internal memory. We assume that $2s + 1$ wing bands are in internal memory. Hence m , the number of vertices in a vertical sweep shape is limited by

$$\left(\left\lceil \frac{2s \cdot (m - 4s) + s}{B} \right\rceil + 1\right) B + \left(\left\lceil \frac{(2s + 1)2s}{B} \right\rceil + 1\right) B \cdot 2 \leq M - cB.$$

This requirement is met, denoting $c' = c + 6$, for

$$m = \left\lfloor \frac{M - c'B - 5s}{2s} \right\rfloor.$$

The number of working bands is $\left\lceil \frac{k_2}{m - 2s} \right\rceil$ and hence the number of vertices in all wing bands is bounded by $\left\lceil \frac{k_2}{m - 2s} \right\rceil \cdot k_1 \cdot 2s$. Since the wings are saved in consecutive blocks, each Block is at most accessed twice during the execution of the algorithm and hence causes at most one compulsory write and read operation. Therefore the number of I/Os is given by

$$\left\lceil \left\lceil \frac{k_2}{m - 2s} \right\rceil \cdot \frac{k_1 \cdot 2s}{B} \right\rceil \cdot 2.$$

Dropping the outer ceiling function adds an error of 2 and dropping the inner ceiling function an error term of

$$\mathcal{O}\left(\frac{k_1}{B}\right).$$

(Error for the last, incomplete working band.) Dropping the $2s$ from the denominator yields an error of

$$\frac{4s \cdot k_1 k_2}{B} \frac{2s}{m^2 - 2ms} = \mathcal{O}\left(\frac{k_1 k_2}{BM^2}\right) .$$

(Estimating the number of working bands - not accounting for outer part of wing bands.) The upper bound can now be approximated by

$$\frac{4s \cdot k_1 k_2}{B} \cdot \frac{2s}{M - c'B - 5s - 2s} ,$$

for which we simplify the denominator to M yielding an error of

$$\frac{8s^2 \cdot k_1 k_2}{B} \cdot \frac{c'B - 7s}{M^2 - c'BM - 7sM} = \mathcal{O}\left(\frac{k_1 k_2}{M^2}\right) .$$

(Separate blocks for core and wing bands, blocks for output.) Hence the upper bound is

$$\frac{8s^2 \cdot k_1 k_2}{BM} + \mathcal{O}\left(\frac{k_1}{B} + \frac{k_1 k_2}{M^2}\right) = \left(8s^2 + \mathcal{O}\left(\frac{M}{k_2} + \frac{B}{M}\right)\right) \cdot \frac{k_1 k_2}{BM} .$$

4.3 Analysis of the Lower Order Terms of the Two Dimensional Block Aligned Diagonal Layout

For the two dimensional block aligned diagonal layout $c = 3$ output blocks have to be reserved. For the input there have to at most

$$\left(\left\lceil \frac{2s(m-2s)+s}{B} \right\rceil + 1\right) B + \left(\left\lceil \frac{(2s+1)s}{B} \right\rceil + 1\right) 2 \cdot B \leq M - cB$$

vertices in internal memory. The first term accounts for the $2s$ core bands and s additional vertices of the $2s+1$.th band. The second term contains the vertices of both wing bands of width s . $2s+1$ diagonals per wing band are kept in memory. Using c' to gather the terms of the ceiling functions and the following $+1$'s we get

$$m = \left\lfloor \frac{M - c'B - 3s}{2s} \right\rfloor .$$

The number of working bands per row can be estimated by $\left\lceil \frac{k_1}{2m-2s} \right\rceil + 1$ as $2m - 2s$ vertices can be evaluated per row of a working band. For each of the two wing bands there are $2s$ vertices per row for which every B vertices there is an I/O. Note that, when the wing bands are ordered bottom up, the upper wing band of a working band is the lower wing band of the next working band. Hence the number of I/Os can be bounded by

$$\left\lceil \left(\left\lceil \frac{k_1}{2m-2s} \right\rceil + 1\right) \frac{2sk_2}{B} \right\rceil \cdot 2 .$$

We are ready to simplify this term. Dropping the outer ceiling functions just adds 2 additional I/Os for the last block of the two last wing bands. Dropping the inner ceiling function and the corresponding +1 yields an error of $\mathcal{O}\left(\frac{k_2}{B}\right)$. (Estimation of the number of working bands.) The current upper bound formula is

$$\frac{k_1}{2m-2s} \cdot \frac{2sk_2}{B} \cdot 2$$

so we next simplify the denominator to $2m$ yielding an error term of

$$\frac{2sk_2}{B} \cdot 2 \cdot k_1 \cdot \frac{1}{2} \cdot \frac{s}{m^2 - ms} = \mathcal{O}\left(\frac{k_1 k_2}{BM^2}\right).$$

(Difference between width of working band and the vertices that can be evaluated in a working band.) Now we substitute the value for m to get

$$\frac{k_1}{2m} \cdot \frac{2sk_2}{B} \cdot 2 = \frac{k_1}{\left\lceil \frac{M-c'B-3s}{2s} \right\rceil} \cdot \frac{sk_2}{B} \cdot 2 \leq k_1 \cdot \frac{sk_2}{B} \cdot 2 \cdot \frac{2s}{M-c'B-5s}.$$

So we simplify the denominator to M yielding an error of

$$k_1 \frac{sk_2}{B} \cdot 2 \cdot 2s \cdot \frac{c'B+5s}{M^2 - c'BM - 5Ms} = \mathcal{O}\left(\frac{k_1 k_2}{M^2}\right).$$

(Blocks for output, wing bands and core bands need separate blocks.) Altogether this yields the upper bound of

$$4s^2 \cdot \frac{k_1 k_2}{BM} + \mathcal{O}\left(\frac{k_2}{B} + \frac{k_1 k_2}{M^2}\right) = \left(\frac{4s^2}{M} + \mathcal{O}\left(\frac{1}{k_1} + \frac{B}{M^2}\right)\right) \cdot \frac{k_1 k_2}{B}.$$

4.4 Analysis of the Lower Order Terms of the Three Dimensional Row Layout

The sweep shape is a square of m vertices per side, lying in an $x_2 x_3$ plane and shifted along x_1 . To need only a constant number of $c = 9$ output blocks the results are saved in a block aligned column layout, as in the two dimensional case. Assuming $B \geq 2s$ one block has to be in internal memory per row of the sweep shape. In the worst case of a block aligned row layout, the $2s$ previous sweep shapes are also needed. As in two dimensions, the vertices in the core band are rearranged in internal memory while the vertices of the wing bands stay in row layout when the input is written back to external memory. The core band is a square band of $m - 4s$ vertices per side. The wing bands consists of $4 \cdot (m - 4s) \cdot 2s + 4 \cdot 2s \cdot 2s = 2s \cdot (m - 2s)$ rows in total. They can be subdivided into two classes. There are four wing bands each when two respectively four working bands overlap. They consists of $(m - 4s) \cdot 2s$ or $2s \cdot 2s$ vertices each. So m has to fulfill

$$m^2 \cdot B + \left(\left\lceil \frac{2s \cdot (m - 4s)^2}{B} \right\rceil + 1\right) \cdot B + 4 \cdot 2s \cdot (m - 2s) \cdot B \leq M - cB. \quad (18)$$

Solving this quadratic equation, adding an additional +1 for dropping the ceiling function, yields

$$m_{1,2} = \frac{16s^2 - 8sB \pm \sqrt{(-16s^2 + 8sB)^2 - 4(B+2s)(32s^3 + 2B - 16s^2B - M + cB)}}{2 \cdot (B+2s)} \dots \frac{1}{1}.$$

Since m only has to suffice the upper bound from (18) we can estimate

$$\begin{aligned} & \frac{16s^2 - 8sB \pm \sqrt{(-16s^2 + 8sB)^2 - 4(B+2s)(32s^3 + 2B - 16s^2B - M + cB)}}{2 \cdot (B+2s)} \geq \\ & \geq \frac{-8sB + \sqrt{4(B+2s)(-32s^3 - 2B + M - cB)}}{2 \cdot (B+2s)} \geq \\ & \geq \frac{\sqrt{M - 32s^3 - 2B - cB} - \frac{4sB}{\sqrt{B+2s}}}{\sqrt{B+2s}}. \end{aligned}$$

So we choose

$$m = \left\lceil \frac{\sqrt{M - 32s^3 - 2B - cB} - \frac{4sB}{\sqrt{B+2s}}}{\sqrt{B+2s}} \right\rceil.$$

For the analysis of the lower order terms we need the asymptotic behavior of m which is $m = \mathcal{O}\left(\sqrt{\frac{M}{B}}\right)$. There are $\left\lceil \frac{k_2}{m-2s} \right\rceil$ working bands in x_2 direction and $\left\lceil \frac{k_3}{m-2s} \right\rceil$ in x_3 direction. The wing bands that take part in two working bands cause one non-compulsory I/Os for each working band they take part in. The wing bands that take part in four working bands cause 6 non-compulsory I/Os in total. We overestimate this by 2 non-compulsory I/Os for every working band they are in. Furthermore we account for another $2m^2$ I/Os per working band to read and write the first and last block of each working band. Altogether the upper bound is

$$\left\lceil \frac{k_2}{m-2s} \right\rceil \left\lceil \frac{k_3}{m-2s} \right\rceil \left(2m^2 + \left(\left\lceil \frac{k_1}{B} \right\rceil + 1 \right) \cdot 4 \cdot (2s \cdot (m-4s) + 2 \cdot 2s \cdot 2s) \right).$$

This is now simplified. First, dropping the first two ceiling functions yields the two error terms

$$\left\lceil \frac{k_3}{m-2s} \right\rceil \cdot \left(2m^2 + \left(\left\lceil \frac{k_1}{B} \right\rceil + 1 \right) \cdot 4 \cdot (2s \cdot (m-4s) + 2 \cdot 2s \cdot 2s) \right) = \mathcal{O}\left(\frac{k_1 k_3}{B}\right)$$

and

$$\frac{k_2}{m-2s} \cdot \left(2m^2 + \left(\left\lceil \frac{k_1}{B} \right\rceil + 1 \right) \cdot 4 \cdot (2s \cdot (m-4s) + 2 \cdot 2s \cdot 2s) \right) = \mathcal{O}\left(\frac{k_1 k_2}{B}\right).$$

(The last, incomplete working band in both directions.) Simplifying the third term to $\frac{k_1}{B}$ adds an error of

$$\frac{k_2}{m-2s} \cdot \frac{k_3}{m-2s} \cdot (2m^2 + 2 \cdot 4 \cdot (2s \cdot (m-4s) + 2 \cdot 2s \cdot 2s)) = \mathcal{O}(k_2 k_3)$$

(Additional I/Os for each row when blocks go over two rows and, lower order, estimating blocks per row.) such that the upper bound reads

$$\frac{k_2}{m-2s} \cdot \frac{k_3}{m-2s} \cdot \frac{k_1}{B} \cdot 4 \cdot (2s \cdot (m-4s) + 2 \cdot 2s \cdot 2s) = \frac{k_2}{m-2s} \cdot \frac{k_3}{m-2s} \cdot \frac{k_1}{B} \cdot 8ms \ .$$

We drop the $-2s$ terms which yields errors of

$$\begin{aligned} k_2 \cdot \frac{2s}{m^2 - 2sm} \cdot \frac{k_3}{m-2s} \cdot \frac{k_1}{B} \cdot 8ms &= \mathcal{O}\left(\frac{k_1 k_2 k_3}{M}\right) \text{ and} \\ \frac{k_2}{m} \cdot k_3 \cdot \frac{2s}{m^2 - 2sm} \cdot \frac{k_1}{B} \cdot 8ms &= \mathcal{O}\left(\frac{k_1 k_2 k_3}{M}\right) . \end{aligned}$$

(Outer wing band cannot be evaluated.) As next step we substitute the value of m and get, using the concavity of the square root,

$$\begin{aligned} \frac{8s \cdot k_1 k_2 k_3}{B} \cdot \frac{1}{m} &= \frac{16s \cdot k_1 k_2 k_3}{B} \cdot \frac{1}{\left[\frac{\sqrt{M-32s^3-2B-cB} - \frac{4sB}{\sqrt{B+2s}}}{\sqrt{B+2s}} \right]} \leq \\ &\leq \frac{8s \cdot k_1 k_2 k_3}{B} \cdot \frac{\sqrt{B} + \sqrt{2s}}{\sqrt{M} - \sqrt{32s^3-2B-cB} - \frac{4sB}{\sqrt{B+2s}} - \sqrt{B+2s}} . \end{aligned}$$

Simplifying the enumerator yields an error of

$$\frac{8s \cdot k_1 k_2 k_3}{B} \cdot \frac{\sqrt{2s}}{\sqrt{M} - \sqrt{32s^3-2B-cB} - \frac{4sB}{\sqrt{B+2s}} - \sqrt{B+2s}} = \mathcal{O}\left(\frac{k_1 k_2 k_3}{B\sqrt{M}}\right)$$

($2s$ previous sweep shapes are needed in memory.) and then simplifying the denominator

$$\begin{aligned} \frac{8s \cdot k_1 k_2 k_3}{B} \cdot \sqrt{B} \cdot \frac{\sqrt{32s^3-2B-cB} + \frac{4sB}{\sqrt{B+2s}} + \sqrt{B+2s}}{M - \sqrt{M} \left(\sqrt{32s^3-2B-cB} - \frac{4sB}{\sqrt{B+2s}} - \sqrt{B+2s} \right)} &= \\ &= \mathcal{O}\left(\frac{k_1 k_2 k_3}{M}\right) . \end{aligned}$$

(Blocks for output and for each part of the core and wing bands.) Hence the number of non-compulsory I/Os is upper bounded by

$$\frac{8s \cdot k_1 k_2 k_3}{\sqrt{B}\sqrt{M}} + \mathcal{O}\left(\frac{k_1 k_2 k_3}{M} + \frac{k_1 k_2 k_3}{B\sqrt{M}}\right) = \frac{k_1 k_2 k_3}{\sqrt{B}\sqrt{M}} \cdot \left(8s + \mathcal{O}\left(\frac{\sqrt{B}}{\sqrt{M}} + \frac{1}{\sqrt{B}}\right) \right) .$$

4.5 Analysis of the Lower Order Terms of the Three Dimensional Block Aligned Column/Pole Layout

The sweep shape is a square of $m \times m$ vertices lying in an x_2x_3 -plane shifted in x_1 direction. As the sweep shape and sweep order are identical to the three dimensional row layout, the size of the wing and core bands are the same and up to $c = 9$ output blocks are needed. The column and pole layout are completely symmetric and the discussion is restricted to the column layout. As the vertices are saved in separate blocks for core bands and wing bands that are shared by different sets of working bands, m has to suffice

$$\begin{aligned} & \left(\left\lceil \frac{2s \cdot (m-4s)^2 + (m-4s) \cdot s}{B} \right\rceil + 1 \right) B + \\ & + \left(\left\lceil \frac{(2s+1) \cdot (m-4s) \cdot 2s}{B} \right\rceil + 1 \right) \cdot B \cdot 4 + \\ & + \left(\left\lceil \frac{(2s+1) \cdot (2s)^2}{B} \right\rceil + 1 \right) \cdot B \cdot 4 \leq M - cB . \end{aligned}$$

The $(m-4s) \cdot s$ part in the second term captures the vertices of the s .th proceeding sweep shape that are also needed in internal memory in addition to $2s$ complete sweep shapes. Using $c' = c + 18$ to take care of the ceiling functions and the corresponding $+1$'s, solving the resulting quadratic equation yields that a solution is given by

$$\begin{aligned} & \frac{-9s + \sqrt{81s^2 - 4 \cdot (2s) \cdot (-M - 20s^2 + c'B)}}{2 \cdot 2s} \geq \\ & \geq \frac{-9s + \sqrt{8s \cdot (M - c'B)}}{4s} = \frac{\sqrt{M - c'B} - \frac{9\sqrt{s}}{\sqrt{2}}}{\sqrt{2s}} . \end{aligned}$$

So we choose

$$m = \left\lceil \frac{\sqrt{M - c'B} - \frac{9\sqrt{s}}{\sqrt{2}}}{\sqrt{2s}} \right\rceil ,$$

which yields $m = \mathcal{O}(\sqrt{M})$ for the analysis of the lower order terms. There are $\left\lceil \frac{k_2}{m-2s} \right\rceil$ working bands in x_2 direction and $\left\lceil \frac{k_3}{m-2s} \right\rceil$ in x_3 direction. The vertices in the wing bands cause non-compulsory I/Os. One non-compulsory I/Os is caused by a vertex in a wing band shared by two working bands for each working band it belongs to. 6 non-compulsory I/Os are caused in total by the vertices in the intersection of four working bands. We estimate this by two non-compulsory I/Os for each participating working band. Hence the upper bound is

$$\begin{aligned} & \left\lceil \frac{k_2}{m-2s} \right\rceil \cdot \left\lceil \frac{k_3}{m-2s} \right\rceil \cdot \\ & \cdot \left(\left(\left\lceil \frac{k_1(m-4s) \cdot 2s}{B} \right\rceil + 1 \right) \cdot 4 + \left(\left\lceil \frac{k_1 \cdot (2s)^2}{B} \right\rceil + 1 \right) \cdot 4 \cdot 2 \right) . \end{aligned}$$

We simplify this term by dropping the ceiling functions and the corresponding +1. This yields the error terms

$$\begin{aligned}
& \left\lceil \frac{k_3}{m-2s} \right\rceil \cdot \left(\left(\left\lceil \frac{k_1(m-4s) \cdot 2s}{B} \right\rceil + 1 \right) \cdot 4 + \left(\left\lceil \frac{k_1 \cdot (2s)^2}{B} \right\rceil + 1 \right) \cdot 4 \cdot 2 \right) = \\
& \qquad \qquad \qquad = \mathcal{O} \left(\frac{k_1 k_3}{B} \right) , \\
& \frac{k_2}{m-2s} \cdot \left(\left(\left\lceil \frac{k_1(m-4s) \cdot 2s}{B} \right\rceil + 1 \right) \cdot 4 + \left(\left\lceil \frac{k_1 \cdot (2s)^2}{B} \right\rceil + 1 \right) \cdot 4 \cdot 2 \right) = \\
& \qquad \qquad \qquad = \mathcal{O} \left(\frac{k_1 k_2}{B} \right) \text{ and} \\
& \frac{k_2}{m-2s} \cdot \frac{k_3}{m-2s} \cdot (2 \cdot 4 + 2 \cdot 4 \cdot 2) = \mathcal{O} \left(\frac{k_2 k_3}{M} \right)
\end{aligned}$$

(The last working band in x_2 and x_3 direction and blocks of wings of working bands that are at the end and beginning of a row.) such that the upper bound has simplified to

$$\begin{aligned}
& \frac{k_2}{m-2s} \cdot \frac{k_3}{m-2s} \cdot \left(\frac{k_1(m-4s) \cdot 2s}{B} \cdot 4 + \frac{k_1 \cdot (2s)^2}{B} \cdot 4 \cdot 2 \right) = \\
& \qquad \qquad \qquad = \frac{k_2}{m-2s} \cdot \frac{k_3}{m-2s} \cdot \frac{k_1}{B} \cdot 8sm .
\end{aligned}$$

(Estimating the number of working bands. Outer part of wing bands cannot be evaluated.) Dropping the two $-2s$ terms yields the error terms

$$\begin{aligned}
& k_2 \cdot \frac{2s}{m^2 - 2sm} \cdot \frac{k_3}{m-2s} \cdot \frac{k_1}{B} \cdot 8sm = \mathcal{O} \left(\frac{k_1 k_2 k_3}{BM} \right) \text{ and} \\
& \frac{k_2}{m} \cdot k_3 \cdot \frac{2s}{m^2 - 2sm} \cdot \frac{k_1}{B} \cdot 8sm = \mathcal{O} \left(\frac{k_1 k_2 k_3}{BM} \right) .
\end{aligned}$$

Substituting the value for m and using the concavity of the square root yields

$$\begin{aligned}
& \frac{k_2}{m} \cdot \frac{k_3}{m} \cdot \frac{k_1}{B} \cdot 8sm = \frac{8s \cdot k_1 k_2 k_3}{B} \cdot \frac{1}{\left\lceil \frac{\sqrt{M-c'B} - \frac{9\sqrt{s}}{\sqrt{2}}}{\sqrt{2s}} \right\rceil} \leq \\
& \leq \frac{8s \cdot k_1 k_2 k_3}{B} \cdot \frac{\sqrt{2s}}{\sqrt{M-c'B} - \frac{9\sqrt{s}}{\sqrt{2}} - \sqrt{2s}} \leq \\
& \leq \frac{8s \cdot k_1 k_2 k_3}{B} \cdot \frac{\sqrt{2s}}{\sqrt{M} - \sqrt{c'B} - \frac{9\sqrt{s}}{\sqrt{2}} - \sqrt{2s}} .
\end{aligned}$$

The enumerator is simplified to \sqrt{M} yielding another error term of

$$\frac{8\sqrt{2} \cdot s^{3/2} \cdot k_1 k_2 k_3}{B} \cdot \frac{\sqrt{c'B} + \frac{9\sqrt{s}}{\sqrt{2}} + \sqrt{2s}}{M + \sqrt{M} \left(-\sqrt{c'B} - \frac{9\sqrt{s}}{\sqrt{2}} - \sqrt{2s} \right)} = \mathcal{O} \left(\frac{k_1 k_2 k_3}{\sqrt{BM}} \right) .$$

(Full blocks for output and each part of the wing bands and the core band.)
Hence the non-compulsory I/Os can be upper bounded by

$$\frac{8\sqrt{2} \cdot s^{3/2} \cdot k_1 k_2 k_3}{B\sqrt{M}} + \mathcal{O}\left(\frac{k_1 k_2 k_3}{\sqrt{BM}}\right) = \frac{k_1 k_2 k_3}{B\sqrt{M}} \cdot \left(8\sqrt{2} \cdot s^{3/2} + \mathcal{O}\left(\frac{\sqrt{B}}{\sqrt{M}}\right)\right) .$$

4.6 Analysis of the Lower Order Terms of the Three Dimensional 2D Block Aligned Diagonal Layout

The sweep shape is a two dimensional ℓ^1 ball of radius m lying in an $x_2 x_3$ -plane and shifted along x_1 . It consists of $2m^2 + 2m + 1$ vertices. Within a sweep shape the vertices are saved in diagonals like in the two dimensional layout but in separate blocks according to core and wing bands that are shared by different working bands. As the wing bands are shared with four other working bands in up to eight different ways, $c = 9$ blocks have to be reserved for the output. Take the intersection of a working band with a $x_2 x_3$ -plane and denote with A , A' and A'' the number of vertices in that intersection that are in the core band, in the wing band shared by two working bands and in the wing bands shared by four working bands respectively. It then holds that $A + 4A' + 4A'' = 2m^2 + 2m + 1$ and $4A' + 4A'' \leq 4 \cdot 2s \cdot (m + 1)$. A is quadratic in m , A' is linear in m and s and A'' solely depends on s and is independent of m . Hence m is limited by

$$\begin{aligned} \left(\left\lceil \frac{2s \cdot A + s \cdot (m + 1)}{B} \right\rceil + 1\right) \cdot B + \left(\left\lceil \frac{(2s + 1) \cdot A'}{B} \right\rceil + 1\right) \cdot B \cdot 4 + \\ + \left(\left\lceil \frac{(2s + 1) \cdot A''}{B} \right\rceil + 1\right) \cdot B \cdot 4 \leq M - cB . \end{aligned}$$

The $s \cdot (m + 1)$ in the first term upper bounds the vertices of the s .th proceeding sweep shape. Estimating the ceiling functions and the corresponding $+1$'s with $c' = c + 18$ and solving the resulting quadratic equation yields an upper bound for m , namely

$$\frac{-13s + \sqrt{(13s)^2 - 4 \cdot (4s) \cdot (11s + c'B - M)}}{2 \cdot 4s} \geq \frac{\sqrt{M - 11s - c'B} - \frac{13\sqrt{s}}{4}}{2 \cdot \sqrt{s}} .$$

Hence we choose

$$m = \left\lfloor \frac{\sqrt{M - 11s - c'B} - \frac{13\sqrt{s}}{4}}{2 \cdot \sqrt{s}} \right\rfloor$$

which has the asymptotic $m = \mathcal{O}(\sqrt{M})$. As next step we estimate the number of working bands. While a quadratic working band is shifted by $m - 2s$ in x_2 - and x_3 -direction, the ℓ^1 ball working band can be shifted by $2m - 3s$ in x_2 direction and m in x_3 direction. Note that the best offset of the working bands varies slightly with s and the core bands overlap for the two dimensional ℓ^1 ball as sweep shape. But this does not affect the leading term of the non-compulsory

I/Os. Again, we account for two non-compulsory I/Os for all vertices belonging to four working bands. The upper bound is then estimated by

$$\left(\left\lceil \frac{k_2}{2m-3s} \right\rceil + 1\right) \cdot \left(\left\lceil \frac{k_3}{m} \right\rceil + 1\right) \cdot \left(\left(\left\lceil \frac{k_1 \cdot A'}{B} \right\rceil + 1\right) \cdot 4 + \left(\left\lceil \frac{k_1 \cdot A''}{B} \right\rceil + 1\right) \cdot 4 \cdot 2\right) .$$

Dropping the ceiling functions and the corresponding +1's yields the error terms

$$\begin{aligned} 2 \cdot \left(\left\lceil \frac{k_3}{m} \right\rceil + 1\right) \cdot \left(\left(\left\lceil \frac{k_1 \cdot A'}{B} \right\rceil + 1\right) \cdot 4 + \left(\left\lceil \frac{k_1 \cdot A''}{B} \right\rceil + 1\right) \cdot 4 \cdot 2\right) &= \mathcal{O}\left(\frac{k_1 k_3}{B}\right) , \\ \frac{k_2}{2m-3s} \cdot 2 \cdot \left(\left(\left\lceil \frac{k_1 \cdot A'}{B} \right\rceil + 1\right) \cdot 4 + \left(\left\lceil \frac{k_1 \cdot A''}{B} \right\rceil + 1\right) \cdot 4 \cdot 2\right) &= \mathcal{O}\left(\frac{k_1 k_2}{B}\right) \text{ and} \\ \frac{k_2}{2m-3s} \cdot \frac{k_3}{m} \cdot (2 \cdot 4 + 2 \cdot 4 \cdot 2) &= \mathcal{O}\left(\frac{k_2 k_3}{M}\right) \end{aligned}$$

(The last working bands in x_2 - and x_3 -direction and the blocks of wing bands at the beginning and end of a row.) such that the upper bound reads

$$\begin{aligned} \frac{k_2}{2m-3s} \cdot \frac{k_3}{m} \cdot \left(\frac{k_1 \cdot A'}{B} \cdot 4 + \frac{k_1 \cdot A''}{B} \cdot 4 \cdot 2\right) &\leq \\ &\leq \frac{k_2}{2m-3s} \cdot \frac{k_3}{m} \cdot \frac{k_1}{B} \cdot (4 \cdot A' + 8 \cdot A'') \leq \\ &\leq \frac{k_2}{2m-3s} \cdot \frac{k_3}{m} \cdot \frac{k_1}{B} \cdot (8s \cdot (m+1) + \mathcal{O}(s^2)) . \end{aligned}$$

We drop the $\mathcal{O}(s^2) = \mathcal{O}(1)$ term, the +1 from $(m+1)$ and simplify the denominator of the first term to $2m$. This yields the error terms

$$\begin{aligned} \frac{k_2}{2m-3s} \cdot \frac{k_3}{m} \cdot \frac{k_1}{B} \cdot \mathcal{O}(s^2) &= \mathcal{O}\left(\frac{k_1 k_2 k_3}{BM}\right) , \\ \frac{k_2}{2m-3s} \cdot \frac{k_3}{m} \cdot \frac{k_1}{B} \cdot 8s &= \mathcal{O}\left(\frac{k_1 k_2 k_3}{BM}\right) \text{ and} \\ k_2 \cdot \frac{3s}{4m^2 - 2m \cdot 3s} \cdot \frac{k_3}{m} \cdot \frac{k_1}{B} \cdot 8sm &= \mathcal{O}\left(\frac{k_1 k_2 k_3}{BM}\right) . \end{aligned}$$

(Overestimating the non-compulsory I/Os caused by the wing bands part of four working bands. Overestimating the vertices in the wing bands. Estimating the number of working bands - outer part of wings cannot be evaluated.) Now substitute the value for m into the upper bound and use the concavity of the square root to get

$$\begin{aligned} \frac{4s \cdot k_1 k_2 k_3}{B} \cdot \frac{1}{m} &= \frac{4s \cdot k_1 k_2 k_3}{B} \cdot \frac{1}{\left\lfloor \frac{\sqrt{M-11s+c'B}-\frac{13\sqrt{s}}{4}}{2\sqrt{s}} \right\rfloor} \leq \\ &\leq \frac{4s \cdot k_1 k_2 k_3}{B} \cdot \frac{2 \cdot \sqrt{s}}{\sqrt{M-11s+c'B}-\frac{13\sqrt{s}}{4}-2\sqrt{s}} . \end{aligned}$$

As last step, simplify the denominator to \sqrt{M} , yielding an error of

$$\frac{8s^{3/2} \cdot k_1 k_2 k_3}{B} \cdot \frac{\sqrt{11s + c'B} + \frac{13\sqrt{s}}{4} + 2 \cdot \sqrt{s}}{M + \sqrt{M} \cdot \left(\sqrt{11s + c'B} - \frac{13\sqrt{s}}{4} - 2 \cdot \sqrt{s} \right)} = \mathcal{O} \left(\frac{k_1 k_2 k_3}{\sqrt{B} M} \right) .$$

(Whole blocks for output and each part of the wing bands and the core band.)
Hence an upper bound for the number of non-compulsory I/Os is

$$\frac{8s^{3/2} \cdot k_1 k_2 k_3}{B\sqrt{M}} + \mathcal{O} \left(\frac{k_1 k_2 k_3}{\sqrt{B} \cdot M} \right) = \frac{k_1 k_2 k_3}{B\sqrt{M}} \cdot \left(8s^{3/2} + \mathcal{O} \left(\frac{\sqrt{B}}{\sqrt{M}} \right) \right) .$$

4.7 Analysis of the Lower Order Terms of the Three Dimensional Hexagonal Aligned Diagonal Layout

Let the algorithm for the hexagonal aligned diagonal layout as described in Sect. 3.2 work on the layout with different blocks for the core bands and wing bands shared by different working bands. Per working band, there are 12 different kinds of wing bands that are accessed so we need $c = 13$ additional output blocks. Consider a working band and its intersection with a plane of normal $(1, 1, 1)$. Of this intersection denote with A the number of vertices that are in the core band, A' the vertices that are in wing bands shared by two working bands and A'' the vertices of the wing bands that are shared by three working bands. We then have $A + 6A' + 6A'' = 3m^2 + 3m + 1$, the full sweep shape. While A depends on m quadratically, A' is linear in m and s and A'' solely depends (quadratically) on s . For a constant $1 \leq d \leq 2$ accounting for the vertices in the $2s + 1$.th sweep shape that have to be in internal memory also, m has to fulfill

$$\left(\left\lceil \frac{2sA + d \cdot ms}{B} \right\rceil + 1 \right) B + \left(\left\lceil \frac{(2s+1)A'}{B} \right\rceil + 1 \right) 6 \cdot B + \left(\left\lceil \frac{(2s+1)A''}{B} \right\rceil + 1 \right) 6 \cdot B \leq M - cB .$$

Dropping the ceiling functions and the corresponding +1s in the constant c' we have to obey

$$\begin{aligned} M - c'B &\geq 2sA + dms + 6(2s+1)A' + 6(2s+1)A'' = \\ &= 2s(3m^2 + 3m + 1) + dms + 6A' + 6A'' . \end{aligned}$$

Since A' is linear in m and s and A'' quadratic in s there are constants d' and d'' such that we need to solve

$$2s(3m^2 + 3m + 1) + d'ms + d''s^2 - M + c'B = 0 .$$

We can restrict ourselves to the positive solution of the quadratic equation.

$$m = \frac{-(6s + d's) + \sqrt{(2s \cdot 3 + d's)^2 - 4 \cdot 6s \cdot (2s + d''s^2 - M + c'B)}}{2 \cdot 6s} .$$

Since we just need an upper bound for m , this equation can be simplified to, using that all involved constants are positive,

$$\begin{aligned}
& \frac{-(6s + d's) + \sqrt{(2s \cdot 3 + d's)^2 - 4 \cdot 6s \cdot (2s + d''s^2 - M + c'B)}}{2 \cdot 6s} \geq \\
& \geq \frac{-(6s + d's) + \sqrt{-4 \cdot 6s \cdot (2s + d''s^2 - M + c'B)}}{2 \cdot 6s} = \\
& = \frac{-\frac{6s+d's}{2\sqrt{6}} + \sqrt{Ms - s(2s + d''s^2 + c'B)}}{\sqrt{6s^2}}.
\end{aligned}$$

Hence we choose

$$m = \left\lceil \frac{\sqrt{Ms - s(2s + d''s^2 + c'B)} - \frac{6s+d's}{2\sqrt{6}}}{\sqrt{6s^2}} \right\rceil.$$

As next step the number of working bands needs to be estimated. Therefore consider the intersection of a working band with an x_1x_2 -plane. The intersection of the working band with the plane consists of $3(3m^2 + 3m + 1)$ vertices. Parts of the wing bands cannot be evaluated but as the wing bands are linear in m and s there is a constant s such that $9m^2 + 9m + 1 - ems \geq 9m^2 - ems$ vertices of these intersection can be evaluated. The width (height) of this intersection, the distance between the leftmost and rightmost (lowermost and uppermost) vertex, is bounded by $7m$. Hence enlarging the grid by $7m$ vertices in positive and negative x_2 - and x_3 -direction ensures that we get a lower bound for the number of working bands in each x_1x_2 -plane when we divide the vertices per plane by the vertices which can be evaluated per working band. Hence the number of working bands is estimated by

$$\frac{(k_2 + 14m)(k_3 + 14m)}{9m^2 - ems}.$$

In the intersection of a working band with an x_1x_2 -plane there are at most $24(m + 1)s$ wing vertices. These divide themselves into wing vertices that are shared by two or three working bands. Denote with C the number of wing vertices that are shared by two working bands and C' the vertices that are shared by three such that $6C + 6C' \leq 24(m + 1)s$. C depends linear on m and s and C'' depends on s quadratically. The number of I/Os is then given by

$$\frac{(k_2 + 14m)(k_3 + 14m)}{9m^2 - ems} \cdot \left(6 \left(\left\lceil \frac{k_1 C}{B} \right\rceil + 1 \right) + 6 \left(\left\lceil \frac{k_1 C'}{B} \right\rceil + 1 \right) \cdot 2 \right).$$

The term including C'' is multiplied by 2 so that we always pay a read and write operation for these vertices. This is too pessimistic since we do not take advantage of the compulsory I/Os.

We can now start simplifying. First drop one of the terms including C' , namely

$$\frac{(k_2 + 14m)(k_3 + 14m)}{9m^2 - ems} \cdot 6 \left(\left\lceil \frac{k_1 C'}{B} \right\rceil + 1 \right) = \mathcal{O} \left(\frac{k_1 k_2 k_3}{BM} \right).$$

(Wing bands shared by three working bands.) Then drop the ceiling functions and the corresponding +1s to get an error of

$$\frac{(k_2 + 14m)(k_3 + 14m)}{9m^2 - ems} \cdot 12 = \mathcal{O}\left(\frac{k_1 k_2}{M}\right).$$

(Blocks at the end and beginning of the wing bands.) The upper bound now reads

$$\begin{aligned} \frac{(k_2 + 14m)(k_3 + 14m)}{9m^2 - ems} \cdot \left(6 \frac{k_1 C}{B} + 6 \frac{k_1 C'}{B}\right) &\leq \\ &\leq \frac{(k_2 + 14m)(k_3 + 14m)}{9m^2 - ems} \cdot \frac{24(m+1)s}{B} \cdot k_1. \end{aligned}$$

Dropping an $24s$ from the numerator of the second term gives an error of

$$\frac{(k_2 + 14m)(k_3 + 14m)}{9m^2 - ems} \cdot \frac{24s}{B} \cdot k_1 = \mathcal{O}\left(\frac{k_1 k_2 k_3}{BM}\right).$$

(Estimation of wing size.) Let us first simplify the denominator to $9m^2$ which yields an error of

$$\frac{(k_2 + 14m)(k_3 + 14m)}{1} \cdot \frac{24ms}{B} \cdot k_1 \cdot \frac{1}{m} \frac{es}{81m^2 - 9ems} = \mathcal{O}\left(\frac{k_1 k_2 k_3}{BM}\right).$$

(Estimation of m and wings of $2s+1$.th layer.) Now the grid sizes can be changed back to k_2 and k_3 gathering the error terms

$$\begin{aligned} 14m \frac{k_3 + 14m}{9m^2} \frac{24ms}{B} k_1 &= \mathcal{O}\left(\frac{k_1 k_3}{B\sqrt{M}}\right) \quad \text{and} \\ \frac{k_2}{9m^2} \cdot 14m \frac{24ms}{B} k_1 &= \mathcal{O}\left(\frac{k_1 k_2}{B\sqrt{M}}\right). \end{aligned}$$

(Estimation of the number of working bands.) We are now ready to substitute the value for m and simplify using the concavity of the square root:

$$\begin{aligned} \frac{k_1 k_2 k_3}{9m^2} \cdot \frac{24ms}{B} &= \frac{24k_1 k_2 k_3}{9B} \frac{1}{\left[\frac{\sqrt{Ms - s(2s + d''s^2 + c'B) - \frac{6s+d's}{2\sqrt{6}}}}{\sqrt{6s^2}} \right]} \leq \\ &\leq \frac{8sk_1 k_2 k_3}{3B} \cdot \frac{\sqrt{6s^2}}{\sqrt{Ms - s(2s + d''s^2 + c'B) - \frac{6s+d's}{2\sqrt{6}} - \sqrt{6s^2}}} \leq \\ &\leq \frac{8sk_1 k_2 k_3}{3B} \cdot \frac{\sqrt{6s^2}}{\sqrt{Ms} - \sqrt{s(2s + d''s^2 + c'B) - \frac{6s+d's}{2\sqrt{6}} - \sqrt{6s^2}}}. \end{aligned}$$

As last step, simplify the denominator to \sqrt{Ms} and collect an error of

$$\begin{aligned} & \frac{8sk_1k_2k_3}{3B} \cdot \sqrt{6s^2} \cdot \\ & \cdot \frac{\sqrt{s(2s + d''s^2 + c'B)} + \frac{6s+d's}{2\sqrt{6}} + \sqrt{6s^2}}{Ms - \sqrt{Ms} \cdot \sqrt{s(2s + d''s^2 + c'B)} - \sqrt{Ms} \cdot \frac{6s+d's}{2\sqrt{6}} - \sqrt{Ms} \cdot \sqrt{6s^2}} = \\ & = \mathcal{O}\left(\frac{k_1k_2k_3}{\sqrt{BM}}\right) . \end{aligned}$$

(Estimation of sweep shape size. Taking into account full blocks for each part of the core and wing bands, difference between working bands and vertices that can be evaluated in a working band.) The error terms simplify to

$$\mathcal{O}\left(\frac{k_1k_2k_3}{BM} + \frac{k_1k_2}{M} + \frac{k_1k_2}{B\sqrt{M}} + \frac{k_1k_2k_3}{\sqrt{BM}}\right) = \mathcal{O}\left(\frac{k_1k_2k_3}{\sqrt{BM}}\right)$$

such that the final upper bound is

$$\frac{8\sqrt{2} \cdot s^{3/2}}{\sqrt{3}} \frac{k_1k_2k_3}{B\sqrt{M}} + \mathcal{O}\left(\frac{k_1k_2k_3}{\sqrt{BM}}\right) = \left(\frac{8\sqrt{2} \cdot s^{3/2}}{\sqrt{3}\sqrt{M}} + \mathcal{O}\left(\frac{\sqrt{B}}{M}\right)\right) \frac{k_1k_2k_3}{B} .$$

4.8 Analysis of the Lower Order Terms of the Arbitrary Dimensional Block Aligned Column Layout

This section analyzes the number of non-compulsory I/Os for the algorithm sweeping an $n - 1$ dimensional hypercube of side lengths m in x_1 -direction in a block aligned column layout. The dimension $n \geq 3$ as well as the stencil size s is held fixed, hence both are $\mathcal{O}(1)$. The number of subsets of working bands that overlap with a fixed working band solely depends on the dimension. Hence the number of blocks needed for the output is $\mathcal{O}(1)$. As input $2s$ sweep shapes of dimension $n - 1$ are needed as well as about $s + 1$ hypercubes of dimension $n - 2$ of the s .th proceeding sweep shape. An additional $\mathcal{O}(B)$ vertices are needed to account for complete blocks. Altogether, m has to suffice

$$2s \cdot m^{n-1} + \mathcal{O}(m^{n-2}) + \mathcal{O}(B) \leq M - \mathcal{O}(B) .$$

In the asymptotic analysis this becomes

$$2s \cdot m^{n-1} + \mathcal{O}(m^{n-2}) = 2s \cdot (m + \mathcal{O}(1))^{n-1} ,$$

such that we can solve

$$m = \sqrt[n-1]{\frac{M - \mathcal{O}(B)}{2s}} - \mathcal{O}(1) .$$

For $i \in \{2, \dots, n\}$ the number of working bands in direction x_i is bounded by $\frac{k_i}{m-2s}$. The working bands extend in x_1 -direction. The $2s$ outermost layers of a

working band are wing bands and $2(n-1)$ sides of a working band are covered by wing bands. For each layer of wing bands and each side of a working band, the intersection with a hyperplane of normal x_1 is a subset of a $n-2$ dimensional hypercube of side length m .

Consider such a hyperplane of normal x_1 for the following discussion. The number of points in the intersection of two adjacent working bands is of order m^{n-2} and the vertices which belong only to these two working bands cause exactly 2 non-compulsory I/Os, one for each working band they are part of. When the intersection three working bands is regarded the number of points is of order m^{n-3} and vertices in only these three working bands cause 4 non-compulsory I/Os. In general, the number of vertices in more than two working bands is of order m^{n-3} or less and the number of I/Os necessary for these vertices solely depends on the number of dimensions. Hence the I/Os caused by vertices in wing bands that are part than more than two working bands are of order $\mathcal{O}\left(\frac{m^{n-3}}{B}\right)$.

We also assume that another $\mathcal{O}(B)$ vertices cause I/Os to account for incomplete blocks of wing bands. Hence an upper bound is given by

$$\prod_{i=2}^n \left\lceil \frac{k_i}{m-2s} \right\rceil \cdot \left(\frac{(2s \cdot 2(n-1) \cdot m^{n-2} + \mathcal{O}(m^{n-3})) \cdot k_1 + \mathcal{O}(B)}{B} \right).$$

This bound is simplified by dropping first the $\mathcal{O}(B)$ and then the $\mathcal{O}(m^{n-3})$ term, yielding errors of

$$\mathcal{O}\left(\frac{\prod_{i=2}^n k_i}{M}\right) \text{ and } \mathcal{O}\left(\frac{\prod_{i=2}^n k_i}{B^{n-1}\sqrt{M^2}}\right).$$

(Error for incomplete blocks in the wing bands. More than two non-compulsory I/Os for vertices that are part of more than two working bands.) Dropping the ceiling functions then, adds an error of at most

$$\mathcal{O}\left(\frac{\prod_{i=1}^{n-1} k_i}{B}\right).$$

(Last, incomplete working band in the last $n-1$ directions.) Simplifying the denominator to m is bounded by

$$\mathcal{O}\left(\frac{\prod_{i=1}^n k_i}{B^{n-1}\sqrt{M^2}}\right).$$

(The outer parts of the wing bands cannot be evaluated.) Substituting m , the upper bound now reads

$$\prod_{i=2}^n \frac{k_i}{m} \cdot \left(\frac{2s \cdot 2(n-1) \cdot m^{n-2} \cdot k_1}{B} \right) = \frac{4s \cdot (n-1) \cdot \prod_{i=1}^n k_i}{B} \cdot \frac{1}{\sqrt[n-1]{\frac{M-\mathcal{O}(B)}{2s}} - \mathcal{O}(1)}.$$

Dropping the $\mathcal{O}(1)$ term adds an error of at most

$$\mathcal{O}\left(\frac{\prod_{i=1}^n k_i}{B^{\frac{1}{n-1} \sqrt{M^2}}}\right)$$

(Vertices needed of the s .th proceeding sweep shape.) and then the concavity of the root can be applied to yield

$$\begin{aligned} & \frac{4s \cdot (n-1) \cdot \prod_{i=1}^n k_i}{B} \cdot \frac{1}{\frac{1}{n-1} \sqrt{\frac{M - \mathcal{O}(B)}{2s}}} \leq \\ & \leq \frac{4s \cdot (n-1) \cdot \prod_{i=1}^n k_i \cdot (2s)^{1/(n-1)}}{B} \cdot \frac{1}{\frac{1}{n-1} \sqrt{M} - \frac{1}{n-1} \sqrt{\mathcal{O}(B)}}. \end{aligned}$$

Finally, dropping the term $\frac{1}{n-1} \sqrt{\mathcal{O}(B)}$ yields an error of

$$\mathcal{O}\left(\frac{\prod_{i=1}^n k_i}{B} \cdot \frac{\frac{1}{n-1} \sqrt{B}}{\frac{1}{n-1} \sqrt{M^2}}\right) = \mathcal{O}\left(\frac{\prod_{i=1}^n k_i}{\frac{1}{n-1} \sqrt{B^{n-2} \cdot M^2}}\right)$$

(Blocks for output, whole blocks for core band and each part of the wing bands.) and the upper bound

$$\frac{4 \cdot 2^{\frac{1}{n-1}} s^{\frac{n}{n-1}} \cdot (n-1) \cdot \prod_{i=1}^n k_i}{B^{\frac{1}{n-1} \sqrt{M}}}.$$

For $n \geq 3$, the last error term is dominant such that the upper bound including error terms is

$$\begin{aligned} & \frac{4 \cdot 2^{\frac{1}{n-1}} s^{\frac{n}{n-1}} \cdot (n-1) \cdot \prod_{i=1}^n k_i}{B^{\frac{1}{n-1} \sqrt{M}}} + \mathcal{O}\left(\frac{\prod_{i=1}^n k_i}{\frac{1}{n-1} \sqrt{B^{n-2} \cdot M^2}}\right) = \\ & = \left(4 \cdot 2^{\frac{1}{n-1}} s^{\frac{n}{n-1}} \cdot (n-1) + \mathcal{O}\left(\frac{\frac{1}{n-1} \sqrt{B}}{\frac{1}{n-1} \sqrt{M}}\right)\right) \cdot \frac{\prod_{i=1}^n k_i}{B^{\frac{1}{n-1} \sqrt{M}}}. \end{aligned}$$

Acknowledgements

The authors like to thank Gero Greiner and Tobias Lieber for helpful discussions during the derivations of the lower and upper bounds.

References

1. Hong, J.W., Kung, H.T.: I/O complexity: The red-blue pebble game. In: Proceedings of STOC '81, New York, NY, USA, ACM (1981) 326–333
2. Aggarwal, A., Vitter, J.S.: The input/output complexity of sorting and related problems. Communications of the ACM **31**(9) (1988) 1116–1127
3. Arge, L., Goodrich, M.T., Nelson, M., Sitchinava, N.: Fundamental parallel algorithms for private-cache chip multiprocessors. In: Proc. of SPAA '08, ACM (2008)

4. Savage, J.E.: Models of Computation: Exploring the Power of Computing. 1st edn. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1997)
5. Bezrukov, S.: Edge isoperimetric problems on graphs. In: Bolyai Math. Series. Volume 7. (1999) 157–197
6. Frumkin, M.A., der Wijngaart, R.F.V.: Minimizing cache misses in scientific computing using isoperimetric bodies. CoRR **cs.PF/0205062** (2002)
7. Frumkin, M.A., Van der Wijngaart, R.F.: Tight bounds on cache use for stencil operations on rectangular grids. J. ACM **49** (May 2002) 434–453
8. Leopold, C.: On optimal locality of linear relaxation. In: Proc. Int. Symp. on Parallel and Distributed Computing and Network. IASTED (2002) 201–206
9. Leopold, C.: An analytical evaluation of tiling for stencil codes with time loop. In: Proc. of the 16th IPDPS, Washington, DC, USA, IEEE Computer Society (2002)
10. Leopold, C.: Tight bounds on capacity misses for 3d stencil codes. In: Proc. of the ICCS. ICCS '02, London, UK, UK, Springer-Verlag (2002) 843–852
11. Leopold, C.: On optimal temporal locality of stencil codes. In: Proceedings of the 2002 ACM symposium on Applied computing. SAC '02, New York, NY, USA, ACM (2002) 948–952
12. Tang, Y., Chowdhury, R.A., Kuzmaul, B.C., Luk, C.K., Leiserson, C.E.: The pochoir stencil compiler. In: Proceedings of SPAA '11, ACM (2011) 117–128
13. Rivera, G., Tseng, C.W.: Tiling optimizations for 3d scientific computations. In: Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM). Supercomputing '00, Washington, DC, USA, IEEE Computer Society (2000)
14. Günther, F., Mehl, M., Pögl, M., Zenger, C.: A cache-aware algorithm for pdes on hierarchical data structures based on space-filling curves. SIAM J. Sci. Comput. **28**(5) (2006) 1634–1650
15. Wellein, G., Hager, G., Zeiser, T., Wittmann, M., Fehske, H.: Efficient temporal blocking for stencil computations by multicore-aware wavefront parallelization. In: Computer Software and Applications Conference, 2009. COMPSAC '09. 33rd Annual IEEE International. Volume 1. (july 2009) 579–586
16. Frigo, M., Strumpen, V.: Cache oblivious stencil computations. In: Proc. of 19th annual ICS '05. ICS '05, ACM (2005) 361–366
17. Frigo, M., Strumpen, V.: The memory behavior of cache oblivious stencil computations. J. Supercomput. **39**(2) (2007) 93–112
18. Strzodka, R., Shaheen, M., Pajak, D., Seidel, H.P.: Cache oblivious parallelograms in iterative stencil computations. In: Proceedings of the 24th ACM International Conference on Supercomputing. ICS '10, New York, NY, USA, ACM (2010) 49–59
19. Zeiser, T., Wellein, G., Nitsure, A., Iglberger, K., Rüde, U., Hager, G.: Introducing a parallel cache oblivious blocking approach for the lattice boltzmann method. In: Progress in Computational Fluid Dynamics. Volume 8 Nos. 1-4. (2008) 179–188
20. Datta, K., Kamil, S., Williams, S., Oliker, L., Shalf, J., Yelick, K.: Optimization and performance modeling of stencil computations on modern microprocessors. SIAM Rev. **51**(1) (2009) 129–159
21. Bollobás, B., Leader, I.: An isoperimetric inequality on the discrete torus. SIAM J. Discret. Math. **3** (January 1990) 32–37
22. Bodlaender, H.L.: A partial k-arborescence of graphs with bounded treewidth. In: J. Algorithms, Springer (1998) 1–16
23. Seymour, P.D., Thomas, R.: Graph searching and a min-max theorem for tree-width. J. Comb. Theory Ser. B **58** (May 1993) 22–33